VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science

Penetration Testing Tool for Web Applications

Nástroj pro penetrační testování webových aplikací

2019 Shanthi Priya Kalirathinam

VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Computer Science

Diploma Thesis Assignment

Student: **Shanthi Priya Kalirathinam**

Study Programme: N2647 Information and Communication Technology

Study Branch: 1801T064 Information and Communication Security

Title: Penetration Testing Tool for Web Applications
Nástroj pro penetrační testování webových aplikací

The thesis language: English

Description:

The thesis will address the vulnerabilities of web applications and automated security audit capabilities. The goal will be to create a comprehensive scanner that automatically detects important information about the web application from the security point of view, finds its vulnerabilities and provides a detailed overview of vulnerabilities and recommendations found. The application should be able, as far as possible, to replace the manual search for vulnerabilities.

Key tasks:

1. Review existing solutions in automated tools for vulnerability detection of web applications
2. Select appropriate test methods and algorithms
3. Implementing the web application scanner
4. Test and compare results with existing tools

References:

[1] P. Prasad, Mastering Modern Web Penetration Testing, Packt Publishing, 2016, ISBN: 978-1785284588
[2] J. A. Ansari, Web Penetration Testing with Kali Linux - Second Edition, Packt Publishing 2015, ISBN: 978-1783988525

Extent and terms of a thesis are specified in directions for its elaboration that are opened to the public on the web sites of the faculty.

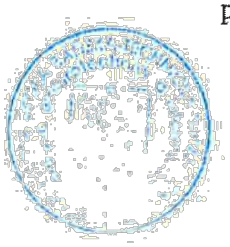
Supervisor: **Ing. Jan Plucar, Ph.D.**

Date of issue: 01.09.2018

Date of submission: 30.04.2019

doc. Ing. Jan Platoš, Ph.D.
Head of Department

prof. Ing. Pavel Brandštetter, CSc.
Dean



I hereby declare that this master's thesis was written by myself. I have quoted all the references I have drawn upon.

Ostrava, 7th June, 2019

I hereby agree to the publishing of the master's thesis as per s. 26, ss. 9 of the Study and Examination Regulations for Master's Degree Programmes at VŠB-Technical University of Ostrava.

Ostrava, 7h June, 2019

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Ing. Jan Plucar, Ph.D. for all the help in connection with the project, for his valuable technical and methodological guidance and support throughout the project period. Without his excellent guidance, this work could not have reached completion.
Secondly, I would like to thank my department (Information and communication security), for providing me certain technical access support for my project and throughout my masters.
Finally, my deepest gratitude to my parents Mr. Kalirathinam Murugan and Mrs. Rajakumari Kalirathinam, and to my sister for their moral support, and blessings.

Abstraktní

Vzhledem k tomu, že se hackeři stávají zkušenějšími a sofistikovanějšími a kybernetické útoky se stávají normou, je důležitější než kdy jindy provádět pravidelné kontroly zranitelnosti a penetrační testování, aby bylo možné identifikovat zranitelná místa a pravidelně zajišťovat fungování kybernetických kontrol. V této práci je diskutován význam a fungování penetračního testování a penetračního testování založeného na webových aplikacích, následuje srovnání a informace o různých testovacích nástrojích a technikách a jejich výhodách a nevýhodách. Další část práce se zaměřuje především na minulý, současný a budoucí stav penetračního testování v počítačových systémech a zabezpečení aplikací, význam nařízení o obecné ochraně údajů (GDPR) a redakčního systému (CMS) následovaného hlavním cílem práce, která vysvětluje stávající řešení v automatizovaných nástrojích pro zjišťování zranitelnosti webové aplikace, jejich techniky, pozitivní a negativní výsledky provedených testů a jejich přednosti a nedostatky. V další části, založené na srovnání různých existujících nástrojů, které vybírají vhodný algoritmus pro diskusi o důležitosti skenování portů, které jsou obvykle zaměřeny na velmi málo stávajících webových aplikací, následující část prakticky demonstruje skenování portů, které poskytují informace týkající se, stav portů pro pochopení informací o službě běžících na serveru. Nakonec bude výsledek experimentu porovnán s existujícími nástroji webové aplikace.

Klíčová slova: penetrační testování, webová aplikace, skenování portů, CMS, zabezpečení, GDPR.

Abstract

As hackers become more skilled and sophisticated and with cyber-attacks becoming the norm, it is more important than ever before to undertake regular vulnerability scans and penetration testing to identify vulnerabilities and ensure on a regular basis that the cyber controls are working. In this thesis the importance and working of penetration testing and web application based penetration testing are discussed, followed by comparison and information's about various testing tools and techniques and their advantages and disadvantages. The next section of the thesis mainly focuses on the past, current and future state of penetration testing in the computer systems and application security, importance of General Data Protection Regulation (GDPR) and Content Management system (CMS) followed by the main goal of the thesis which explains the existing solutions in automated tools for vulnerability detection of web application their techniques, positive and negative results of the conducted tests and their merits and demerits. In the next section, based on the comparison of various existing tools selecting appropriate algorithm for discussing the importance of scanning the ports which are usually focused in very few existing web application tools, the following section practically demonstrate the scanning of ports which gives information regarding, the state of ports to understand the service information running on the server. Finally the result of the experiment will be compared with the existing web application tools.

Keywords: penetration testing, web application, port scan, CMS, security, GDPR.

Contents

List of Symbols and Abbreviations

CMS - Content Management System

WCMS - Web Content Management System

DAST - Dynamic Application Security Testing

SAST - Static Application Security Testing

XSS - Cross-site scripting

CSRF - Cross-site request forgery

LDAP - Lightweight Directory Access Protocol

List of Figures

List of Tables

1. INTRODUCTION

Penetration testing is widely used to help ensure the security of web applications. It discovers vulnerabilities by simulating attacks from malicious users on a target application. Penetration testing looks at vulnerabilities and will try and exploit them. Penetration testing usually stops because of budget issues and sometimes they are also terminated once the goal of the Penetration testing process have been achieved, but this may leave other vulnerabilities unchecked.

The testing is often stopped when the objective is achieved, this means there can be other exploitable vulnerabilities not tested. Based on the study penetration testing tools cannot find all vulnerabilities. Only vulnerabilities that cause results that can be monitored for are found. Therefore, they do a poor job at finding vulnerabilities in all access providing areas that are vulnerable.

According to various available researches it is clear that most of the penetration testing tools are not giving detailed information about network configuration which is one of the most important factor when it comes to gaining network access. Based on publicly available information and penetration tests performed, most of the penetration testing tools are not giving detailed information about ports status and detailed information, which is one of the most important factor when it comes to hacking.

Penetration tests can discover vulnerabilities or potential breaches at an earlier which will be very useful to avoid breach disclosure. Beyond just identifying vulnerabilities prior to real-world exploitation, penetration tests help teams prioritize security fixes based on the severity and impact of different findings. The main difference between penetration testing and some other cyber security testing services is that it relies on the premise that it takes a hacker to know a hacker.

Penetration testing is conducted by ethical hackers, who are best suited to create an event most likely to resemble a real hacking attack and data breach. By finding smaller threats, penetration testers can observe certain patterns on how these can eventually lead to bigger threats and thus prevent a more wide-ranging attack. It is the best way to simulate real cyber security attack that the company would likely face. It provides actionable information about the tested system, but also about the hackers.

Testing the security of web applications with automated penetration testing tools produces relatively quick and easy results. Testing the security of web applications is very important. There are several ways to do this. One such way is by using penetration testing tools. These tools test the security by performing an attack, without malicious payload (i.e. they will not delete parts of the web application or the database it uses), against the web application that should be tested. Results of these attacks are monitored by the tool to see which succeed. There are a lot of penetration testing tools available, with different qualities.

2. STATE OF ART

Penetration testing first became a concept in the 1960s. The burgeoning tech industry realized then that having multiple users on one system, as had become the norm, posed an inherent risk to the system's security. Based on the research this realization gave rise to what became known as “Tiger Teams”, the first of these worked for the government and military. In 1971, the US Air Force ordered security testing of time-shared computer systems. The idea of actually testing systems to ensure their integrity arose with the major security networks such as the RAND Corporation that first identified this now major threat to internet communication. The RAND Corporation, in cooperation with the Advanced Research Projects Agency (ARPA) in the USA, produced a seminal report, generally called The Willis Report after its lead author. The report discussed the security problem and proposed policy and technical considerations that even today lay the groundwork for security measures. From this report, government and business began to put together teams that would try to find vulnerabilities in computer networks and systems to protect the computer systems from unethical hacking or penetration.

With reference to the article An Information Security Pioneer (ISSN: 1558-4046) written by Mr.Eugene Spafford and published by IEEE**,** one of the early pioneers in penetration testing development was James P. Anderson. In his 1972 report, Anderson outlined a series of definitive steps that tiger teams could take to test systems for their ability to be penetrated and compromised. Anderson’s approach included first identifying vulnerability and designing an attack on it, and then finding the weakness in the attack itself and ways to neutralize its threat. This fundamental method is still in use today.

In the 1970s and 1980s, research into how to create a secure system was still novel. Anderson’s 1980 publication that showed how to design a program to monitor the use of a computer system to identify unusual use that might signal hacker activity is so simple that any savvy computer user today would readily understand how it works and be able to point to any number of ways to get around it. Still, the work at the time was groundbreaking, and many of its methods form part of standard system protection today.

Another system that was developed and used by a broad range of government, military and corporate entities was Multics (Multiplexed Information and Computing Service). It may be the earliest of computer systems, operating in some form or other from 1965 to 2000, and arguably still operating today. Honeywell eventually purchased it and serviced education, government, and industry. The key development that came from Multics was that it delivered secure computing service to users in remote locations, a radical development for the period. Fundamental designs from Multics are still in use even today in other operating systems such as UNIX.

The Multics security system was so good it became the first and for many years the only operating system awarded a B2 rating by the US government. Still, in 1974, the US Air Force conducted an ethical hack on its Multics system, one of the earliest known white hat attacks in the USA, and plenty of vulnerabilities were revealed. Regardless, Multics is still considered to be one of the most secure systems in the world, in part because all of its security features are part of the standard product, rather than supplementary or add on features.

As a result, application designers had to ensure that their product met the access design security clearances if they wanted their products to work with the Multics system. Today, though, when security features can be optional, and often are, applications may not be able to meet such requirements, leaving individual computer users vulnerable to hacking.

Later in 1984, the US Navy got involved with ethical hacking action when a team of Navy Seals worked to evaluate how easily terrorists could access different naval bases. Around the same time, the US government was starting to come down on illegal hackers. One result of this process was the Computer Fraud and Abuse

Act, which specified that particular ethical hacking techniques were only allowed under a contract between hacker and client organization.

In the 1990s, a security administrator tool for analyzing networks (SATAN) became available. The name scandalized, and developers added a feature allowing it to be reconfigured to SANTA, a testament to the perhaps natural mischievous nature of penetration testers and hackers. The tool allowed administrators to run a series of tests on their own networks to help identify areas of possible vulnerability and created a report along with a tutorial explaining what issues might arise.

SATAN is no longer in development and has been replaced by other tools such as nmap and Nessus. With reference to Farmer, Dan; Wietse Venema *(1993)* in 1995, Dan Farmer of Sun Microsystems and Wietse Venema of the Eindhoven University of Technology released a paper entitled "Improving the Security of Your Site by Breaking Into It.",(Sun Microsystems. Eindhoven University of Technology. Retrieved 30 May 2014). Farmer and Venema described the emergence of the "uebercracker," a hacker who had evolved beyond the ordinary and had learned to develop his own hacking programs.

This person can discover bugs in the most advanced security systems and can get in and out of a system without leaving a trace. They showed rather than told the importance of a system owner's looking at his or her own system in the way a hacker might, thus laying the groundwork for contemporary penetration testing. In the same year, John Patrick of IBM termed this process "ethical hacking."

In The 2000s after the turn of the new millennium, penetration testing finally began to solidify as a discipline. In 2003, the Open Web Application Security Project (OWASP) published its Testing Guide, which delineated the industry's first set of best practices.

Six years later, the Penetration Testing Execution Standard (PTES) offered providers of penetration testing services with a set of common practices. In 2013, calculations revealed that spending on enterprise security had exceeded $6 billion. Skilled ethical hackers now have a marketplace that desperately needs what they are able to do, so long as employers continue to realize how important it is to stay secure against the smartest attackers.

Today, the available options for penetration testing are highly specialized and numerous. Many systems include tools for a range of security testing of the operating system. One example among many is the Kali Linux, used in digital forensics and penetration testing.

It contains eight standard security tools including Nmap, Aircrack-ng, Kismet, Wireshark, Metasploit Framework, Burp Suite and John the Ripper. That a single system would contain so many penetration testing tools demonstrates how much more sophisticated today's technology has become and how many ways ingenious hackers are discovering to create mischief in shared computing environments, especially the Internet.

The statistics on threats posed by hackers are sobering. A recent RAND report suggests that in one year as many as 65 million people in the USA alone have had their personal data breached in some way or other, and that cyber-crime generates billions of dollars in revenue each year. As well, the very tools created by those who work to secure cyber information can also be used to exploit it.

Today, on-demand penetration testing is one of the latest methods to test a network system for ways it could be breached and information accessed. This hybrid approach to testing a network combines the manual and real-time attempts by ethical hackers to breach the system security alongside automated tools that run checks on the system. Together, this approach is thought to offer a broader and more rigorous security review.

The method has evolved to include subscription-based services. This approach allows smaller companies that might not be able to afford either the wide array of penetration testing tools or the person with the expertise to operate them all to hire an expert to check their system as needed. Since many system-wide checks are run semi-annually, this approach can be a cost effective one, especially for smaller organizations.

With the advancement in technologies in this current generation it is not only important to test and secure government and big organizations, it is also important to secure small firms. After researches most of the testing tool available today are performing similar operations, but only few tools are concentrating more on port configuration.

Hence, I am trying to implement a tool, which scans the website for default and finger directories followed by performing SQL injection to make sure if there is any possibility for SQL vulnerability in the website and finally by scanning the ports, we can obtain the status of the ports which will be very helpful to make sure that, there is no access doors opened for the hackers to enter into the network. Since the importance of GDPR is also being discussed as one of the main objectives of this thesis and as GDPR proposes the importance of safeguarding user information, I am trying to implement a password protection tool which helps to create a strong password thus, the login credentials will not be easily hacked.

3. OBJECTIVES

The main goal of the thesis are to achieve the following:

- To discuss the importance of penetration testing in web based application.
- To practically demonstrate the importance of network configuration in web application.
- Determine the severity of a potential vulnerability due to misconfiguration in web application.
- To illustrate the importance of GDPR.
- To illustrate the need to ensure Web Content Management Systems (WCMS) are kept update.

3.1 Motivation

As the use of web applications is increasing among a number of different industries, many companies turn to online applications to promote their services. Companies see the great advantages with web applications such as convenience, low costs and little need of additional hardware or software configuration.

Meanwhile, the threats against web applications are scaling up where the attacker is not in need of much experience or knowledge to hack a poorly secured web application as the service easily can be accessed over the Internet. Penetration testing is a method used to estimate the security of a computer system, network or web application.

With reference to the Several studies regarding security penetration testing it is clear that even though the penetration testing tools are relatively fast and can detect certain security vulnerabilities easily, the major preference of the testing is to solve the top critical vulnerabilities and to achieve the objective that has been planned before testing process, hence less vulnerable threats are mostly neglected to be solved at earlier stage, which might become critical in the later days and can cause severe vulnerability issues.

According to various available researches that have been conducted by testers most of the penetration testing tools are not giving detailed information about ports which is one of the most important factor when it comes to hacking. As hackers use various ways to enter the network, ports play a significant role, if an

important port is opened it is way easier for the hacker to enter the network so, it is very important to check the status and working of ports on a regular basics to keep the network secure.

3.2 Background and Overview

Penetration testing is a type of security testing that is used to test the insecurity of an application. It is conducted to find the security risk which might be present in the system. If a system is not secured, then any attacker can disrupt or take authorized access to that system. Security risk is normally an accidental error that occurs while developing and implementing the software.

For example, configuration errors, design errors, and software bugs, etc.

The importance of Penetration testing are as follows:

- It identifies a simulation environment i.e., how an intruder may attack the system through white hat attack.
- It helps to find weak areas where an intruder can attack to gain access to the computer's features and data.
- It supports to avoid black hat attack and protects the original data.
- It estimates the magnitude of the attack on potential business.
- It provides evidence to suggest, why it is important to increase investments in security aspect of technology.

The type of penetration testing normally depends on the scope and the organizational wants and requirements. This chapter discusses different types of Penetration testing, also known as Pen Testing.

3.2.1 Types of Penetration testing

The types of Penetration testing are as follows:

- Black Box Penetration Testing
- White Box Penetration Testing
- Grey Box Penetration Testing

Black Box Penetration Testing

In a real world Cyber-attack, the hacker probably will not know all of the ins and outs of the IT infrastructure of a corporation. Because of this, he or she will launch an all-out, brute force attack against the IT infrastructure, in the hopes of trying to find a vulnerability or weakness on which they latch onto.

In other words, in this type of Pen Test, there is no information given to the tester about the internal workings of the particular Web Application, nor about its source code or software architecture. As a result, this particular type of test can take a very long time to complete, so very often, the tester will rely upon the use of automated processes to completely uncover the weaknesses and vulnerabilities. This type of test is also referred to as the "trial and error" approach.

White Box Penetration Testing

White Box Penetration Testing, also known as "Clear Box Testing," the tester has full knowledge and access to both the source code and software architecture of the Web Application. Because of this, a White Box Test can be accomplished in a much quicker time frame when compared to a Black Box Test. The other advantage of this is that a much more thorough Pen Test can be completed.

But, this approach also has its set of disadvantages. First, since a tester has complete knowledge, it could take more time to decide on what to focus specifically on regarding system and component testing and analysis. Second, to conduct this type of test, more sophisticated tools are required such as that of software code analyzers and debuggers.

Gray Box Penetration Testing

As the name implies, this type of test is a combination of both the Black Box and the White Box Test. In other words, the penetration tester only has a partial knowledge of the internal workings of the Web Applications. This is often restricted to just getting access to the software code and system architecture diagrams.

With the Gray Box Test, both manual and automated testing processes can be utilized. Because of this approach, a pen tester can focus their main efforts focus on those areas of the Web Application, which he or she knows the most about, and from there, and from there, exploit any weaknesses or vulnerabilities. With this particular method, there is a higher probability that more hard to find "security holes" will also be discovered as well.

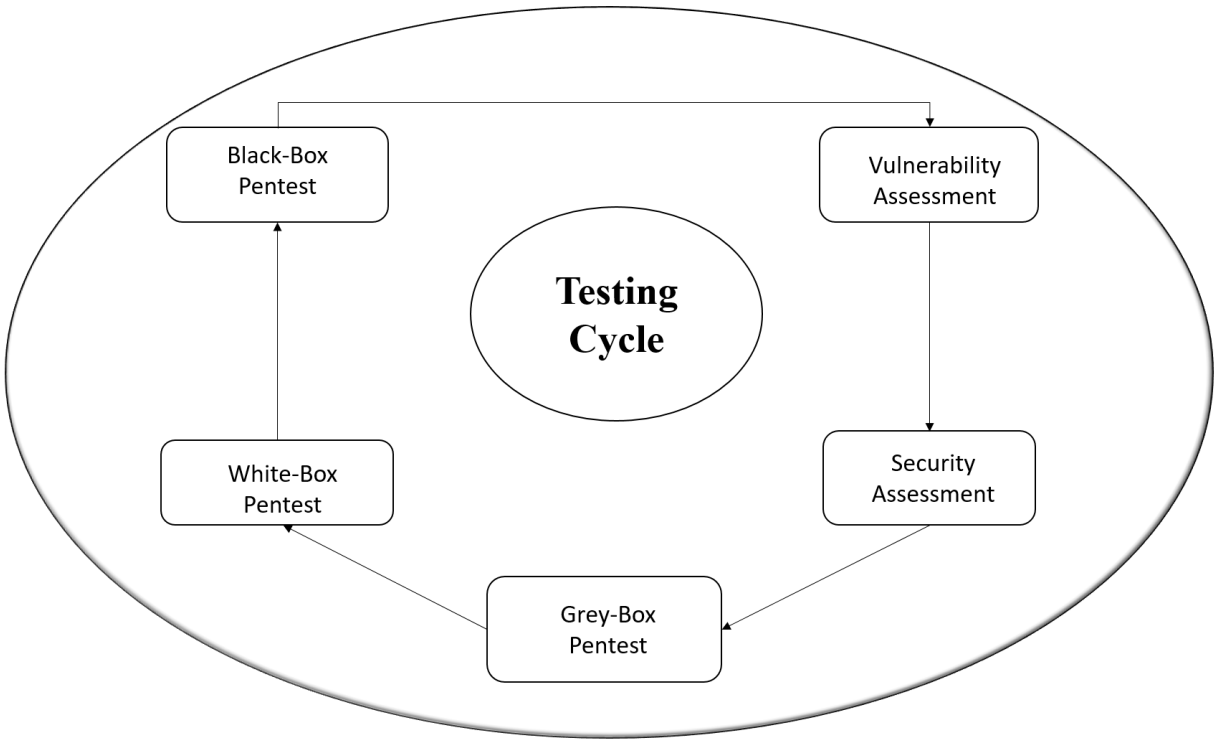


Figure 1. Penetration Testing Cycle Layout

3.2.1.1 Comparison between Black box, White box and Grey box Penetration testing

Black Box Penetration Testing	White Box Penetration Testing	Grey Box Penetration Testing
• Tester has no information of the inner workings of the IT product under test. • Black box testing techniques can be performed by developers, user groups and testers. • As the task of the tester is blocked by absence of data in regards to the auxiliary code inside the product, the approach towards black box automated testing includes an experimentation technique. • The sample space for test inputs is entirely enormous and the biggest among all. • A fast outlining of test cases is conceivable. • Automated black box testing is not appropriate for calculation testing. • Black box testing methodologies is the slightest time depleting type of testing. • Black box security testing has its application in testing versatility of the product against malicious code assaults. • Hard to discover hidden errors.	• Testers have full knowledge of inner programming rationale of the IT product under test. • Execution of automated white box testing is the selective domain of the testing and improvement group. • Since absence of definite learning of the codes is not an issue, WBT can continue by confirming the framework limits and information areas inherent in the app. • Test space for test contributions to be utilized for making test cases is the smallest. • Outlining of test cases takes quite a more time. • Various types of white box testing are viewed as ideal for calculation testing. • White box testing in software engineering is the most tedious type of testing. • Not utilized for testing product strength against viral attacks. • Disclosure of concealed flaws is easy to execute. • WBT also called clear box testing, open box testing, auxiliary testing and logic-driven testing.	• Both black box testing and white box testing are utilized (Mainly for database testing) • In gray box testing techniques inner programming is partially known. • Somewhat knowledge of internal working of application is known. • Gray box testing non-intrusive also known as translucent testing. • Performed by end clients and furthermore by testers and developers. • Gray box testing done on the premise of abnormal state database outlines and information stream chart. • Incompletely tedious and exhaustive. • Not suited to calculation testing. • Information areas and internal limits can be tested if known.

• Black box testing in software engineering also called as opaque testing and specifications based testing.		

Table 1. Comparison of Black box, White box and Grey box Penetration testing

3.2.2 Five Phases of Penetration Testing

The five phases of penetration testing refer to each primary step in the process of conducting a penetration test.

Phase 1 - Reconnaissance

During the reconnaissance phase, an attacker attempts to gather as much information as possible about a target before launching any attacks. Reconnaissance target range could include employees, networks, systems, and even third-parties. Sometimes, information available online about a target could uncover weak entry points depending on what an attacker finds. An attacker would like to know more about the target as far as who they are and what it is there field of work. There are two different types of reconnaissance techniques that an attacker can use. They are passive reconnaissance and active reconnaissance. Passive reconnaissance would involve grabbing information without directly interacting with the target. For example, searching through public records (Google hacking), dumpster diving, news releases, and job posts. Active reconnaissance involves directly interacting with the target. For example, contacting an employee and asking questions about the organization or using non-intrusive network scanning techniques.

Phase 2 - Scanning

Scanning is known as the pre-attack phase when the attacker scans the organization's network to dive deeper into the systems found on the network while looking for valuable data and services. Scanning includes the use of tools such as port scanners, ping tools, vulnerability scanners, and network mappers. For example, using ICMP ping to check if a system is alive on the network. After finding out that the system is alive, Nmap is used to detect open ports and what services are running on the system. Basically, looking for any vulnerabilities that could be exploited in order to gain access. An organization may have an intrusion detection in place on their network, so it is important to learn and use evasion techniques to avoid any alerts.

Phase 3 - Gaining Access

At this point, an attacker has gained access to a system or application and continues to search for more information, for example, an administrator account to escalate privileges. An attacker could gain access to a system or application through password cracking, session hijacking, social engineering, etc. Once the attacker gains access to an administrator account or finds a way to create one, that attacker now has complete control over the system and other systems on the network could be compromised. However, an attacker could still do damage to a system without administrator privileges since there are tools that can run without the need for them.

Phase 4 - Maintaining Access

Maintaining access refers to the phase when an attacker tries to retain some or full control of the system. This could be done by installing rootkits, trojans, creating an admin account, and the use of other backdoor tools. Whatever route an attacker chooses, he or she would be able to upload, download, and manipulate files at any moment the system is online. And most of these things are done remotely. It would be very difficult for a system administrator or user to detect these tools as they are known to run in stealth mode. The attacker could also add the compromised system to a botnet in order to have even more control over the system, which could be used later on for attacks on another target.

Phase 5 - Covering Tracks

The final phase of the attacker is covering their tracks. It is important that the attacker either manipulate or delete logs in order to avoid detection and more importantly prosecution. For example, there are tools that can be used to clear system and application logs from the system, or the attacker can choose to do it manually. An attacker would make sure that trojans and rootkits remain hidden and hide data using stenography techniques in order to avoid detection by an antivirus software. Spoofing IP address is another evasion technique used by an attacker to throw a security team in the wrong direction. Of course, the last thing that any attacker wants is to get caught and lose control of the system.

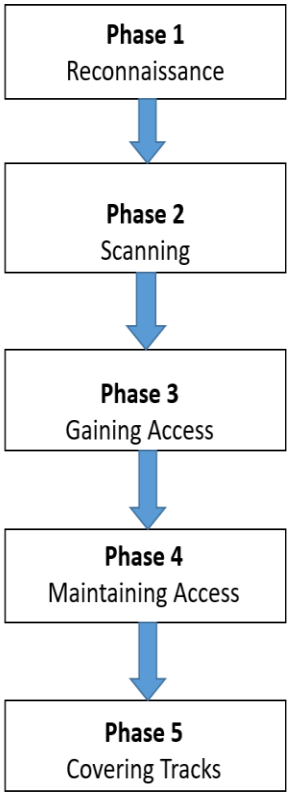


Figure 2: Five Phases of Penetration Testing

The following chapter mainly focuses on Web penetration process. Web penetration testing is typically a manual endeavor, with skilled penetration testers seeking to exploit weaknesses in software in the same way

that cyber criminals do. As a manual procedure, penetration testing can be expensive and time-consuming. Where automated web scanning techniques can return results within hours, it may take weeks to complete web penetration testing on a single application.

3.2.3 Penetration testing limitations

Undertaking a series of penetration tests will help test some of the security arrangements and identify improvements, but it is not a panacea for all ills. For example, a penetration test:

- Covers just the target application, infrastructure or environment that has been selected.
- Focuses on the exposures in technical infrastructure, so it is not intended to cover all the ways in which critical or sensitive information can leak out of organization.
- Plays only a small part (despite often including social engineering tests) in reviewing the people element (often the most important element of an organization's defense system).
- Is only a snapshot of a system at a point in time.
- Can be limited by legal or commercial considerations, limiting the breadth or depth of a test.
- May not uncover all security weaknesses, for example due to a restricted scope or inadequate testing.
- Provides results that are often technical in nature and need to be interpreted in a business context.

4. WEB APPLICATION PENETRATION TESTING

This section lists the penetration testing mechanism, implementation and importance of penetration testing in various fields such as CSP and GDPR. A security test is a method of evaluating the security of a computer system or network by methodically validating and verifying the effectiveness of application security controls. A web application security test focuses only on evaluating the security of a web application. The process involves an active analysis of the application for any weaknesses, technical flaws, or vulnerabilities. Any security issues that are found will be presented to the system owner, together with an assessment of the impact, a proposal for mitigation or a technical solution.

Vulnerability
A vulnerability is a flaw or weakness in a system design, implementation, operation or management that could be exploited to compromise the system's security objectives.

Threat
A threat is anything (a malicious external attacker, an internal user, a system instability, etc) that may harm the assets owned by an application (resources of value, such as the data in a database or in the file system) by exploiting a vulnerability.

Test
A test is an action to demonstrate that an application meets the security requirements of its stakeholders.

4.1 Penetration Testing Mechanism

The six phases of penetration testing are critical for the successful planning and execution of a penetration test. Learn more about each of the phases of penetration testing in the points below.

4.1.1 Pre-Engagement Interactions

During this pre-phase, a penetration testing company will outline the logistics of the test, expectations, legal implications, objectives and goals the customer would like to achieve.

During the Pre-Engagement phase, the penetration testers should work with the company to fully understand any risks, the organizational culture, and the best penetration testing strategy for the organization to perform a white box, black box, or gray box penetration test. It's at this stage when the planning occurs along with aligning the goals to specific penetration testing outcomes.

4.1.2 Reconnaissance or Open Source Intelligence (OSINT) Gathering

Reconnaissance or Open Source Intelligence (OSINT) gathering is an important first step in penetration testing. A penetration tester works on gathering as much intelligence on the organization and the potential targets for exploit.

Depending on which type of penetration test agreed upon, the penetration tester may have varying degrees of information about the organization or may need to identify critical information on their own to uncover vulnerabilities and entry points in the environment.

Common intelligence gathering techniques include:

- Search engine queries
- Domain name searches/WHOIS lookups
- Social Engineering
- Tax Records
- Internet Foot printing – email addresses, usernames, social networks,
- Internal Foot printing –Ping sweeps, port scanning, reverse DNS, packet sniffing
- Dumpster Diving
- Tailgating

A penetration tester uses an exhaustive checklist for finding open entry points and vulnerabilities within the organization.

4.1.3 Threat Modeling & Vulnerability Identification

During the threat modeling and vulnerability identification phase, the tester identifies targets and maps the attack vectors. Any information gathered during the Reconnaissance phase is used to inform the method of attack during the penetration test.

The most common areas a penetration tester will map and identify include:

- Business assets – identify and categorize high-value assets
  - Employee data
  - Customer data
  - Technical data
- Threats – identify and categorize internal and external threats
  - Internal threats – Management, employees, vendors, etc.
  - External threats – Ports, Network Protocols, Web Applications, Network Traffic, etc.

A penetration tester will often use a vulnerability scanner to complete a discovery and inventory on the security risks posed by identified vulnerabilities. Then the penetration tester will validate if the vulnerability is exploitable. The list of vulnerabilities is shared at the end of the penetration test exercise during the reporting phase.

4.1.4 Exploitation

With a map of all possible vulnerabilities and entry points, the penetration tester begins to test the exploits found within a network, applications, and data. The goal is for the ethical hacker is to see exactly how far they can get into the environment, identify high-value targets, and avoid any detection.

If a scope is established initially, then the penetration tester will only go as far as determined by the guidelines agreed upon during the initial scoping. For example, to define the scope to not penetration test cloud services or avoid a zero-day attack simulation.

Some of the standard exploit tactics include:

- Web Application Attacks
- Network Attacks
- Memory-based attacks
- Wi-Fi attacks
- Zero-Day Angle
- Physical Attacks
- Social engineering

The ethical hacker will also review and document how vulnerabilities are exploited as well as explain the techniques and tactics used to obtain access to high-value targets. Lastly, during the exploitation phase, the ethical hacker should explain with clarity what the results were from the exploit on high-value targets.

4.1.5 Post-Exploitation, Risk Analysis & Recommendations

After the exploitation phase is complete, the goal is to document the methods used to gain access to the organization's valuable information. The penetration tester should be able to determine the value of the compromised systems and any value associated with the sensitive data captured.

Some penetration testers are unable to quantify the impact of accessing data or are unable to provide recommendations on how to remediate the vulnerabilities within the environment. Sanitized penetration testing report that clearly shows recommendations for fixing security holes and vulnerabilities should be presented.

Once the penetration testing recommendations are complete, the tester should clean up the environment, reconfigure any access he/she obtained to penetrate the environment, and prevent future unauthorized access into the system through whatever means necessary.

Typical cleanup activities include:

- Removing any executables, scripts, and temporary files from compromised systems.
- Reconfiguring settings back to the original parameters prior to the penetration test.
- Eliminating any rootkits installed in the environment.
- Removing any user accounts created to connect to the compromised system.

4.1.6 Reporting

Reporting is often regarded as the most critical aspect of a penetration test. It's where it will obtain written recommendations from the penetration testing company and have an opportunity to review the findings from the report with the ethical hackers.

The findings and detailed explanations from the report will offer insights and opportunities to significantly improve security posture. The report should show exactly how entry points were discovered from the OSINT and Threat Modeling phase as well as how to remediate the security issues found during the Exploitation phase.

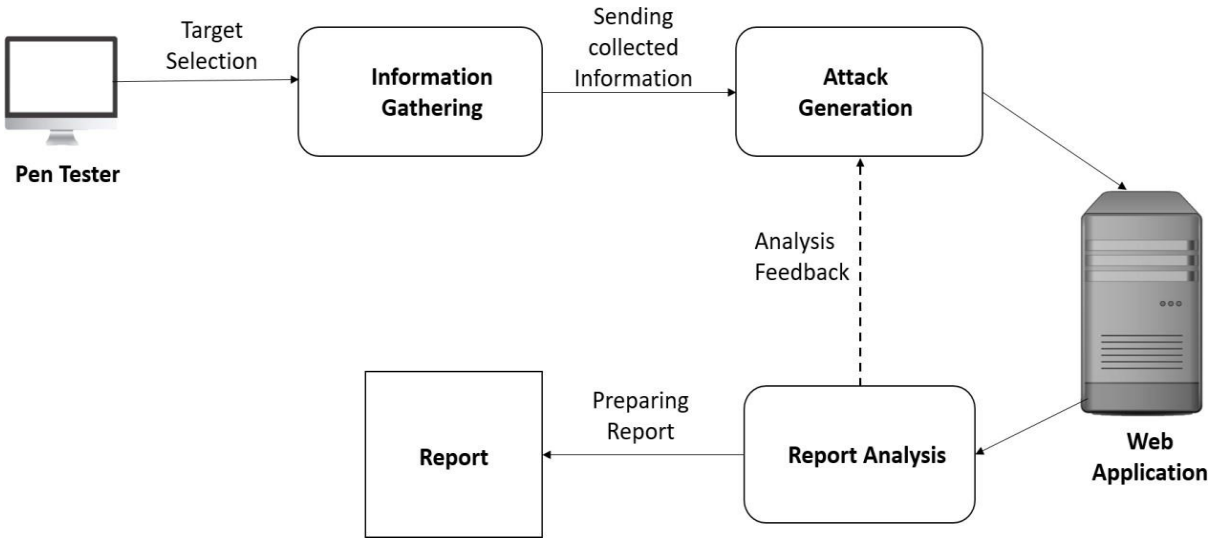


Figure 3: Penetration Testing Mechanism

4.2 Types of Web Application Security Testing

Dynamic Application Security Testing (DAST): A DAST approach involves looking for vulnerabilities in a web app that an attacker could try to exploit. This testing method works to find which vulnerabilities an attacker could target and how they could break into the system from the outside. Dynamic application security testing tools don't require access to the application's original source code, so testing with DAST can be done quickly and frequently.

Static Application Security Testing (SAST): SAST has a more inside-out approach, meaning that unlike DAST, it looks for vulnerabilities in the web application source code. Since it requires access to the application source code, SAST can offer a snapshot in real time of the web application security.

Application Penetration Testing: Application penetration testing involves the human element. A security professional will try to imitate how an attacker might break into a web app using both their personal security know-how and a variety of penetration testing tools to find exploitable flaws. Also web application penetration testing services can be outsourced to a third party if there is no need of resources in-house.

Security vulnerabilities are a weak point of any system at the design and implementation levels, and WCMS are not an exception.

4.3 Penetration testing for Content Management and Content Security Policy

A secure application ensures authentication, confidentiality, integrity, and availability. Web based Content Management System are becoming popular in today's internet, and have become a target for attackers.

If a Web based Content Management System has security vulnerabilities, it may become inaccessible. Usual attack consequences are confidential data destruction, data modification, and misuse of a web-server for illegal activities, and Denial-of-Service (DoS), among others.

4.3.1 Content Management Systems

There are vulnerabilities in Web Content Management Systems (WCMS) which are often overlooked, such as stored and reflected cross-site scripting attacks. The purpose of WCMS's is to provide an easy alternative method for website designers and developers to create interactive web site content that is easy to both manage and administer. Various benefits can be gained from using a WCMS over custom developed websites. One of which is an "increased level of security."

Content Management Systems (CMS) like Drupal, WordPress, Magento and Joomla are extremely popular and ideal for editing content. However, these systems are also very vulnerable to hackers, provided the security is not regularly maintained and checked.

In addition to the standard installation, different plugins, themes and custom modules are often installed. Precisely these plugins and custom code are prone to security breaches. So, it is hard to understand the number of vulnerabilities that are indirectly installed through plugins written by random programmers.

Even standard installations contain errors and vulnerabilities exploited by hackers. This stresses the importance of always updating CMS to the latest version, otherwise hackers can easily hack the website.

Web Security Scan offers security scans and extensive penetration tests to thoroughly check the Content Management System, including all plugins, themes and other custom modules for vulnerabilities and security risks. This way it will help to understand how well the website is being protected.

4.3.2 Content Security Policy

The entire web application security model is built on the idea of "Same Origin Policy" which restricts the content of one domain being accessed by another domain. By subverting this policy, the attackers have introduced various attacks like cross-site scripting and other data injection attacks. In order to mitigate such attacks, the concept of "content – security – policy" was brought into picture.

"Content Security Policy" is a declarative policy that allows the application developers to inform the client about the sources from which the application can load the resources. Content security policy does not help as a first line of defense against code injection attacks, but it can be best used as defense-in-depth to reduce the harm caused by such attacks. To take advantage of this policy, the application developers need to make use of "Content-Security-Policy" HTTP header.

Background

Content security policy header was originally developed by Mozilla Foundation. Experimental implementations of this header in various browsers was done by names like X-Webkit-CSP in chrome , X-Content-Security-Policy in browsers like Mozilla, SeaMonkey, etc. "Content-Security-Policy" is the standard header name proposed by the W3C document.

Working

The content-security-policy header is sent in the server response. On encountering this header in the response, the client judges from where or from where not the content should be loaded in that page. This is done by the browser by using the directives and the values present in the CSP response header. CSP

header is applicable per page. Hence, it is advised to add this header for every response.

Another important aspect of implementing penetration testing is in GDPR. The Cyber Security Breaches Survey 2017 found that virtually all UK businesses covered by the survey are exposed to cyber security risks. 61% hold personal data on their customers electronically. Worryingly, 46% of all UK businesses identified at least one cyber security breach or attack in the past 12 months. The survey also found that businesses that hold electronic personal data on customers are more likely than average to have experienced a breach.

A penetration test aims to determine whether and how an attacker can gain unauthorized access to assets that affect the fundamental security of the system. It provides real-world security testing of the security controls that are believed to be in place and functioning effectively.

4.4 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is focused on the personal data of citizens within the European Union. GDPR is often viewed as having two primary goals within the EU and beyond:

- To define the online rights of EU citizens.
- To regulate the handling of EU citizen's personal data.

The EU has changed its data protection rules. These new rules are called the General Data Protection Regulation (or GDPR). A key goal of GDPR is to provide EU citizens with more control over their own data. The advertisement of products online after searching for that same product just a day prior is mainly because of the data collected the day prior. GDPR sets out to make this optional & completely up to the individual.

Businesses should prepare the significant changes that go well beyond an upgrade. Many deals signed will be covered by the new law. The General Data Protection Regulation (GDPR) is focused on the personal data of citizens within the European Union.

The General Data Protection Regulation (GDPR) protects the personal data of EU citizens. If the company is handling the personal data of EU citizens, no matter where the company is, it need to take some important steps to ensure that data is correctly controlled, processed, maintained, retained, and secured. Failure to comply with GDPR can lead to penalties of much as €20 Million or 4% of the annual gross revenue (whichever is higher).

Under GDPR, individuals have the following list of rights:

- **To be informed:** Before data is collected on individuals, the individual must knowingly give consent.
- **Access:** If requested, companies must provide individual's access to what data has been collected about individual and how that data is being used.
- **Rectification:** If data is old or incorrect, individuals have the right to have the data corrected.
- **Erasure:** If an individual is no longer a customer, or has withdrawn approval of data collection, then an individual has the right to have data fully deleted.
- **To Restrict Processing:** An individual has the right to request their data not be used for any processing, but the data does not have to be deleted.
- **Data Portability:** An individual has the right to have data moved from one company to another.
- **To Object:** Individuals have the right to immediately stop their data from being used in direct marketing.

- **Rights related to automated decision making including profiling:** Individuals have the right to know if automated decision making is being used in a way that can impact them.

In addition to individual rights, GDPR also aims to regulate how personal data such as name, email, address, etc. is handled. Companies must be transparent in what information is collected and how it will be used. As it is noticed by the multiple privacy policy updates that have been received by the company. Companies are not only explaining how data is captured, stored, and used, but also secured. While GDPR is primarily viewed as a privacy standard, there are also underlying roots in security.

4.4.1 GDPR and Penetration Testing

Does GDPR need a penetration test?

It may initially seem as if GDPR solely applies to EU companies, but any company handling the data of EU citizens must be in compliance. In addition to this, there is reason to believe that companies may choose to treat all consumers under GDPR guidelines. The reason for this is that it may logistically make sense to handle all customers with this new standard instead of handling customer data in the EU different from consumers in the rest of the world.

Managing and maintaining compliance requires a security infrastructure that can monitor and control the use and movement of data, identify the users who are using the data, restrict access to only those users who need to access it, and to render the data unintelligible in the event that it is accessed by an unauthorized user. The GDPR recommends to assess applications and critical infrastructure for security vulnerabilities and that the effectiveness of the security controls are tested regularly, services such as penetration testing and regular vulnerability assessments would help meet this recommendation. And with breach reports being legally required (no later than 72 hours).

There are some underlying affects to security professionals. A key development in GDPR is the requirements around breach announcements. By examining large breaches such as Equifax, companies tend to know far before the effected consumers find out. With GDPR, the new standard is 72 hours from the discovery of a breach. Security professionals will have more cause to stay on top of analysis & internal communication of security concerns. In addition to this, GDPR stresses the importance of what is referred to as "privacy-by-design." As SaaS platforms & web applications are developed, security and privacy must be front of mind. If the development teams overlooks security in exchange for sooner release dates, it can quickly find them in trouble. As part of this, on-going penetration testing & security assessments of such applications will be key in ensuring privacy-by-design.

4.4.2 Importance of GDPR and Penetration Testing

According to Article 32 of GDPR:

"(d) a process for regularly testing, assessing and evaluating the effectiveness of technical and organizational measures for ensuring the security of the processing".

Article 32 requires that the controller and the processor shall implement appropriate technical and organizational measures to ensure a level of security appropriate to the risk, including;

(a) The pseudonymisation and encryption of personal data.

(b) The ability to ensure the ongoing confidentiality, integrity, availability and resilience of processing systems and services.

(c) The ability to restore the availability and access to personal data in a timely manner in the event of a physical or technical incident.

(d) A process for regularly testing, assessing and evaluating the effectiveness of technical and organizational measures for ensuring the security of the processing.

This is little vague as it doesn't specifically define what must be tested on a regular basis. A strong rule of thumb is that any systems or applications that touch personal data. If Article 32 isn't enough of a reason to understand why penetration testing is an important security factor of GDPR, the mandatory breach disclosure should be enough of an incentive. Penetration tests can discover vulnerabilities or potential breaches before anyone else.

GDPR will create the perfect reason to have regular penetration tests, but when it comes down to it penetration tests are helpful to any team. Beyond just identifying vulnerabilities prior to real-world exploitation, penetration tests help teams prioritize security fixes based on the severity and impact of different findings.

GDPR penetration testing is also one of the main concerns when it comes to Cloud Security, While GDPR has caused panic among IT environments worldwide, the complications around data security in cloud environments is even more complex. AWS for example, has GDPR compliance supported through many of its services, but doesn't reduce the financial penalties in the event of a data breach (regardless of who's at fault or how it happened).This greater impact raises the risk and concern around AWS penetration testing and the proper configuration and handling of environments.

GDPR Penetration Testing To Protect Personal Data

Always look to protect the personal data that are being held. The network that stores or processes this data forms the target for penetration testing. By testing how easy it is to access that network and the personal data held or processed there. The tester should confirm that all routes into the network are secure and only allow the right people access. These routes include external (Internet facing) networks and any unrelated internal company networks. So overall the main aims of GDPR is to make sure that personal data is secure and access is limited to only the people that need it.

GDPR Basics.

Make sure to know where the Personal Data is: By finding the networks that hold and process data. Document this as it forms the scope of our GDPR network.

Remove what is not needed: Delete the data that are not required. It will reduce risk and will reduce the GDPR compliance burden.

Lock access down: Not just from the Internet. Internal networks should be secured. This is often called segmentation. It controls access to networks and data. This is important if the confidential employee records should not be viewed by everyone.

Get staff working securely: Being aware of phishing emails and online fraud attempt. Getting staff educated in these tasks.

Test: Make sure to test the network and people for resilience. Awareness training for staff. Penetration testing and vulnerability testing for networks. These tasks support best practice standards for ISO 27001 and should be done periodically

Repeat: Revisiting GDPR basics. Refining and strengthening controls.

5. COMMON VULNERABILITIES

This section lists the vulnerabilities that are being tested in most of the available tools

Cross-site scripting

Cross-site scripting (also known as **XSS**) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. It allows an attacker to circumvent the same origin policy, which is designed to segregate different websites from each other. Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, to carry out any actions that the user is able to perform, and to access any of the user's data. If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the application's functionality and data.

There are three different types of XSS: stored XSS, reflected XSS and DOM based XSS. The differences between these types are that, for stored XSS the attacker's code is stored on the web server (e.g. the guest book of the example above), whereas for reflected XSS the attacker's code is added to a link to the web application (e.g. in a GET parameter) and the attacker has to trick a user into clicking on the link. Such a link would look like http://www.example.com/index.php?input=<script>alert(‘XSS’);</script>. For Dom based XSS the attacker's code is not injected in the web application, instead the attacker uses existing JavaScript code on the target page to write text (e.g. <script>alert(‘XSS’);</script> on the page. To test for this vulnerability, a penetration testing tool should try to input HTML code in the inputs on a web application. After this, the tools would have to search for the code that was inputted, to see if it is present.

SQL Injection

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior. In some situations, an attacker can escalate an SQL injection attack to compromise the underlying server or other back-end infrastructure, or perform a denial-of-service attack.
The following example is a query used by a user login. This query is usually like "SELECT * FROM users WHERE username='entered username' AND password='entered password' ". If an attacker enters the string x' OR '1'='1 in both the username and the password field the query becomes "SELECT * FROM users WHERE username='x' OR '1'='1' AND password='x' OR '1'='1' ". Because '1' is always equal to '1', this query is true for all records in the database. There are two different types of SQL injection: blind SQL injection and "normal" SQL injection. The difference between these two types is that for "normal" SQL injection the server shows an error message when the SQL queries syntax in incorrect, for blind SQL injection this error message is not shown. Instead the attacker will see a generic error message or page. "Normal" SQL injection can be tested for by entering characters like quotes to create a query with an incorrect syntax and search the page for error messages about it. Blind SQL injection cannot be detected this way, instead the attacker has to enter SQL commands like sleep or statements that are always true or false. For instance trying both strings ' AND '1'='1 and ' AND '1'='2 will likely produce different results if the page is vulnerable to SQL injection.

XPath Injection

XPath Injection is an attack technique used to exploit applications that construct XPath (XML Path Language) queries from user-supplied input to query or navigate XML documents. It can be used directly by an application to query an XML document, as part of a larger operation such as applying an XSLT transformation to an XML document, or applying an XQuery to an XML document.

The syntax of XPath bears some resemblance to an SQL query, and indeed, it is possible to form SQL-like queries on an XML document using XPath. XPath injection vulnerabilities arise when user-controllable data is incorporated into XPath queries in an unsafe manner. An attacker can supply crafted input to break out of the data context in which their input appears and interfere with the structure of the surrounding query. Depending on the purpose for which the vulnerable query is being used, an attacker may be able to exploit an XPath injection flaw to read sensitive application data or interfere with application logic.

Similar to SQL injection, there are two types of XPath injection: "normal" XPath injection and blind XPath injection. The difference between these two types of XPath injection is that for blind XPath injection the attacker has no knowledge about the structure of the XML document and the application does not provide useful error messages. Testing for XPath injection is also similar to SQL injection. The first step would be to insert a quote in an input field to see if it produces an error message. For blind XPath injection data is injected to create a query that always produces true or false.

Cross site request forgery

Cross site request forgery (CSRF), also known as XSRF, Sea Surf or Session Riding, is an attack vector that tricks a web browser into executing an unwanted action in an application to which a user is logged in. A successful CSRF attack can be devastating for both the business and user. It can result in damaged client relationships, unauthorized fund transfers, changed passwords and data theft—including stolen session cookies. CSRFs are typically conducted using malicious social engineering, such as an email or link that tricks the victim into sending a forged request to a server. As the unsuspecting user is authenticated by their application at the time of the attack, it’s impossible to distinguish a legitimate request from a forged one.

For example an attacker might post the following HTML on a website or send it in an HTML email <img src= http://www.bank.com/transfer_money?amount=10000&target_account=1234567>.If the user is authenticated at his bank website (at http://www.bank.com) when this link is loaded it would transfer 10000 from the user's account to bank account number 12345]. Testing for this attack is pretty similar to testing for XSS, the tool will have to check if it can inject a link that may have effect on another web application (e.g. the link of the example) into the web application that is being tested.

HTTP response splitting

HTTP Response Splitting (CWE-113 - Improper Neutralization of CRLF Sequences in HTTP Headers), also known as CRLF is an attack where the attacker can control the data that is used in an HTTP response header and enters a newline in this data. For example, if a web application uses a redirect via a GET parameter (e.g. http://www.example.com/index.php?page=somepage.html), this redirect is sent via the HTTP headers to the browser (the "Location" header). An attacker can append a newline to the value of the GET parameter and add his own headers. This way an attacker can add a "normal" response header and can cause text to appear on the web application this way. A penetration testing tool would have to enter a newline, followed by a HTTP header, in inputs that may be present in the HTTP response header. The tool will then have to check whether the server returns the data that is inputted in the header by the attacker.

LDAP injection

LDAP injection (CWE-90 - Improper Neutralization of Special Elements used in an LDAP Query) is an attack where the attacker inputs LDAP statements that are executed by the server. There are two types of LDAP injection: "normal" LDAP injection and blind LDAP injection. Just like with SQL injection and XPath injection, the difference between these two types is that with blind LDAP injection no error messages are shown. To test for LDAP injection a penetration testing tool should enter (parts of) LDAP statements in inputs for normal LDAP injection. For blind LDAP injection true or false questions should be entered in the inputs.

Remote file inclusion

Remote file inclusion is equal to local file inclusion, except for that the file that is included is a file from a different server than the one the web application is running on. An example of this vulnerability is the same as for local file inclusion. However, instead of changing the file name parameter to a local file, the attacker should enter a path to a remote file. Testing for this vulnerability is also similar to local file inclusion. However, instead of a path to a local file a path to a remote file should be used (e.g. http://www.something.com/index.html).

6. PENETRATION TESTING TOOLS AND WEB APPLICATION PENETRATION TESTING TOOLS

Web security testing tools are useful in proactively detecting application vulnerabilities and safeguarding websites against attacks.

Nmap

Nmap (Network Mapper) is a free and open source utility for network discovery and security auditing.
While systems and network administrators can use it for tasks such as network discovery and inventory, pentesters can similarly employ Nmap for reconnaissance and scanning, getting basic information such as what hosts are available on the network; what services (application name and version) those hosts are offering; what operating systems (and OS versions) they are running; what type of packet filters/firewalls are in use and several of other characteristics.

Nmap also includes a scripting module, so it is not limited to gathering basic information. Aside from network discovery, it can also perform vulnerability and backdoor detection, and even execute exploitations.

Advantages	Disadvantages
• NMap provides a and a thorough network "sweep" that allows to quickly map out exactly what's on the network. • NMap is highly configurable. The scanned choices are very good in most instances, but using various switches and options, can create a very specific scan and get the needed results. • NMap is easy to use. Even a new administrator will be able to use the graphical version (Zenmap) with efficiency right away.	• Running stealthier scans would be a bonus. Current scans are little confusing. • Scans run fast, which sometimes can make it look like a system is being attacked. There is a slow, comprehensive scan option, though. • NMap scripts are written in Lua, which is not a mainstream language.

Table 2: Pros and cons of Nmap

Metasploit

Metasploit is an amazing tool for penetration testing. In fact, Metasploit is a framework and not a specific application, meaning it is possible to build custom tools for specific tasks. It comes in several versions (both free and paid), available for both Windows and Linux.
Metasploit is quite simple to use and was specifically designed to aid penetration testers. The common steps for exploiting any target are:

- Selecting and configuring the exploit to be targeted
- Selecting and configuring the payload that will be used
- Selecting and configuring the encoding schema that will be used for trying to evade intrusion detection systems (IPSs)
- Executing the exploit

Advantages	Disadvantages
• Vulnerability exploiting • Tool integration such as with Nmap • Very intuitive interface and searching	• More robust menus • Better plugin inter-operation

Table 3: Pros and cons of Metasploit

Nessus

Nessus is an excellent vulnerability scanner. It provides comprehensive detection, including the ability to identify vulnerabilities, configuration issues and even malware on web applications.

Nessus is fast and accurate, and even though it is not designed for executing exploitations, it can be of greater value for penetration testers during the reconnaissance and scanning phases. It provides detailed target information that can be used by other tools (such as Metasploit) for exploitation.

Advantages	Disadvantages
• They have many plugins available for the various vulnerabilities that are out there. • The ability to scan static and dynamic asset lists is good. • The ability to schedule recurring scan jobs is helpful and aids in the scanning of systems. Integration with ticketing system such as *ServiceNow* is also good.	• The website does not have the published or revised date of their plugins. • The UI could be improved so that queries didn't take so long. It would be nice if there was an easy way to purge old data associated with particular IPs.

Table 4: Pros and cons of Nessus

Burp Suite

Burp Suite is an integrated platform used for testing the security of web applications. It contains several tools that work seamlessly together, supporting the entire testing process. Burp can perform the initial mapping and analysis of an application attack surface, and goes as far as finding and exploiting security vulnerabilities. It contains the following components:

- **Intercepting proxy:** For inspecting and modifying traffic between the browser and the target application
- **Application-aware spider:** For crawling content and functionality
- **Advanced web application scanner:** For automating the detection of numerous types of vulnerabilities
- **Intruder tool:** For performing powerful customized attacks to find and exploit unusual vulnerabilities
- **Repeater tool:** For manipulating and resending individual requests
- **Sequencer tool:** For testing the randomness of session tokens

Burp also allows for the creation of plugins for performing complex and customized tasks. It is easy to use, highly customizable and contains numerous powerful features that can help the most experienced penetration testers. In other words, it is an excellent tool for performing web application security assessments.

Advantages	Disadvantages
• Burp Suite is fairly quick to perform an attack on a website. I have found it very thorough for the time it takes to run an attack.	• Burp Suite is not a tool that a complete security novice will get much out of. By knowing the basics of application security to be able to properly use the tool.
• Burp Suite can spider a website very quickly and it usually finds most of the hyperlinks on a website. Once it has spidered a website, it allows to not attack any page it found during the scan.	• Burp Suite can, at times, take a very long time to completely attack a website. I have found that some websites are still being attacked after a few hours. This is usually due to errors being thrown during the attack process and Burp Suite has determined that too many errors have been thrown it will stop attempting the test that was throwing the errors.
• Burp Suite allow to easily log into a website as the first step in spidering and attacking. This is useful for us since most of our websites require a login before we can scan the internal pages of a website.	• Burp Suite is constantly being updated. I find that I have to install a new release about two or three times a month. I know this should be considered a good thing, and it can be, but sometimes I am afraid that an update might break the tool.

Table 5: Pros and cons of Burp Suite

6.1 Common Open Source Security Testing Tools for Web Applications

The testing tool helps to identify the security lapse in the web applications. Its primary function is to perform the functional testing of an application and find the vulnerabilities that could lead to data leak or hacking, without accessing the source code.

A. Wapiti

- Wapiti is one of the efficient web application security testing tools that allow to assess the security of the web applications. It performs a blackbox scan and injects payloads to check if a script is vulnerable. It supports both GET and POST HTTP attack methods. It detects vulnerabilities like file Disclosure, file inclusion, cross Site Scripting (XSS), weak .htaccess configuration, database injection, Command Execution detection, CRLF Injection, XXE injection, potentially dangerous files.

- Wapiti is a command-line application. So, without deep knowledge of commands it's hard to work with Wapiti tool.

B. W3af

W3af is a popular web application security testing framework. Developed using Python, It is a web application audit and attack framework that is effective against over 200 vulnerabilities, including SQL injection and Cross-Site Scripting. It has a GUI with expert tools which can be used to send HTTP request and cluster HTTP responses. If a website is protected, it can use authentication

modules to scan them. Output can be logged into a console, a file or sent via email.

- Blind SQL injection vulnerability
- Buffer overflow vulnerability
- Multiple CORS misconfigurations
- Insecure DAV configurations
- CSRF vulnerability and much more

Unlike Wapiti it is available in both GUI and console interface, it is easy to understand and allows to authenticate the website through the authentication modules.

C. SQLMap

SQLMap is a popular open source web application security testing tool. It detects SQL injection vulnerability in a website database. It can be used on a wide range of databases and supports six kinds of SQL injection techniques: time-based blind, boolean-based blind, error-based, UNION query, stacked queries and out-of-band. It can directly connect to the database without using an SQL injection and has great database fingerprinting and enumeration features. SQLMap supports a large number of database services, including MySQL, Oracle, PostgreSQL, Microsoft SQL Server etc.

Based on the detailed study about various testing tools, the tools that have discussed above are some of the common security tools used by most of the security testers. There are some more top Web security tools which are commonly used. Some of these tools will tested in the following chapter.

7. EXPERIMENT

This section will focus on the penetration testing tools, explaining the inner workings of the tools according to the claims on the tools' website. Here, the tools and the vulnerabilities they can detect will be described.

Some of the key parts that will be discussed in the following section are as follows:

- Focusing on the tested tools inner workings and features of the tools according to the claims on the tools' website
- Comparing results based on the respective tools website claims.
- Comparing the results of each tool with one another.
- Discussing merits and demerits from obtained result.

7.1 Test Setup

An example website is created for the testing purpose. The main goal is to test the website with various existing tool both offline and online tools by performing attack in the created website. After first test is performed, vulnerabilities on the website will be fixed and website will be tested again.

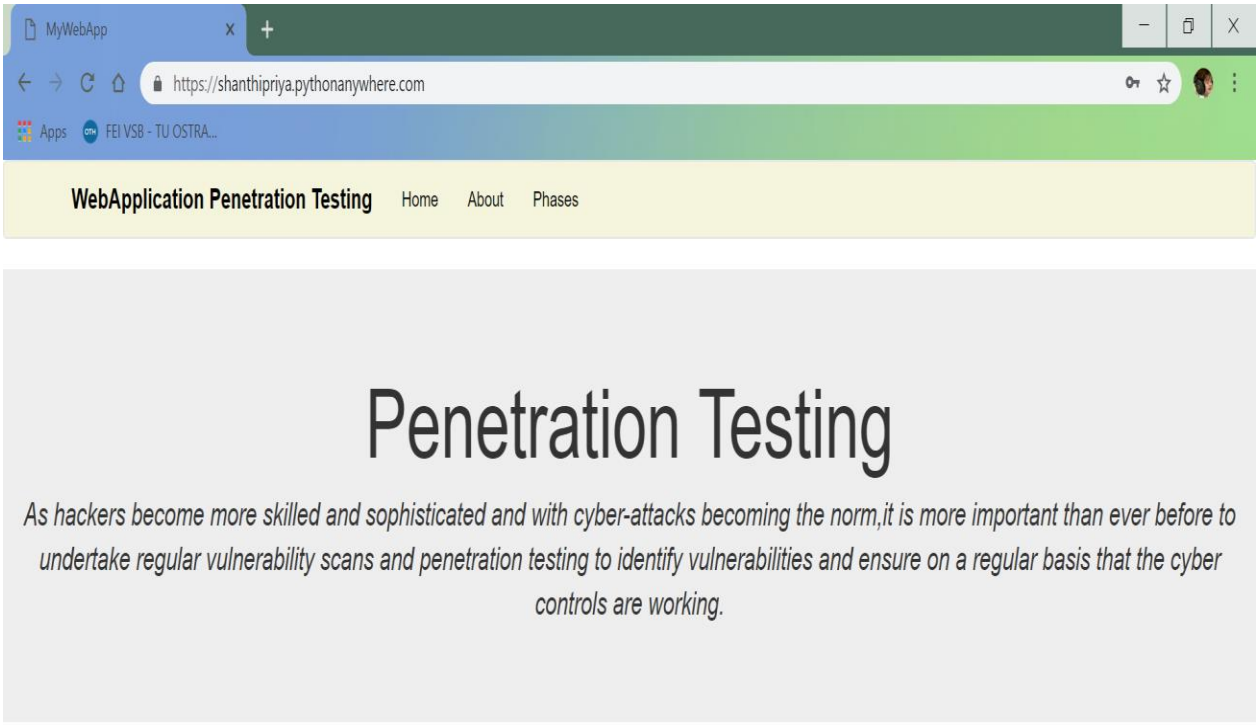


Figure 4. Testing Website

7.2 Testing existing Web application Testing Tools

For experiment purpose, I am performing a Reflected server-side cross-site scripting in my flask web application, by writing user input directly to a web page which, allows for a cross-site scripting vulnerability.

The following example shows the way to render the given name back to the page. This method is vulnerable as `Pname` is not escaped, leaving the page vulnerable to cross-site scripting attacks.

```
from flask import Flask, request, make_response
app = Flask(__name__)

@app.route('/xss')
def xss():
Pname = request.args.get('name', '')
return make_response("Name " + Pname)
```

Now, the XSS affected website will be tested in various tools and the results will be compared.

A. Zed Attack Proxy

Popularly known as ZAP, the Zed Attack Proxy is an open source, developed by OWASP. Supported by Windows, Unix/Linux and Mac OS, ZAP enables to find a variety of security vulnerabilities in web apps, even during the development and testing phase. This testing tool is easy to use, even for a beginner. The key features of Zap are:

- Automatic Scanner
- Authentication support
- AJAX spiders
- Dynamic SSL certificates
- Forced Browsing
- Intercepting Proxy
- Web Socket Support
- Plug-n-hack support
- REST-based API and much more.

Result

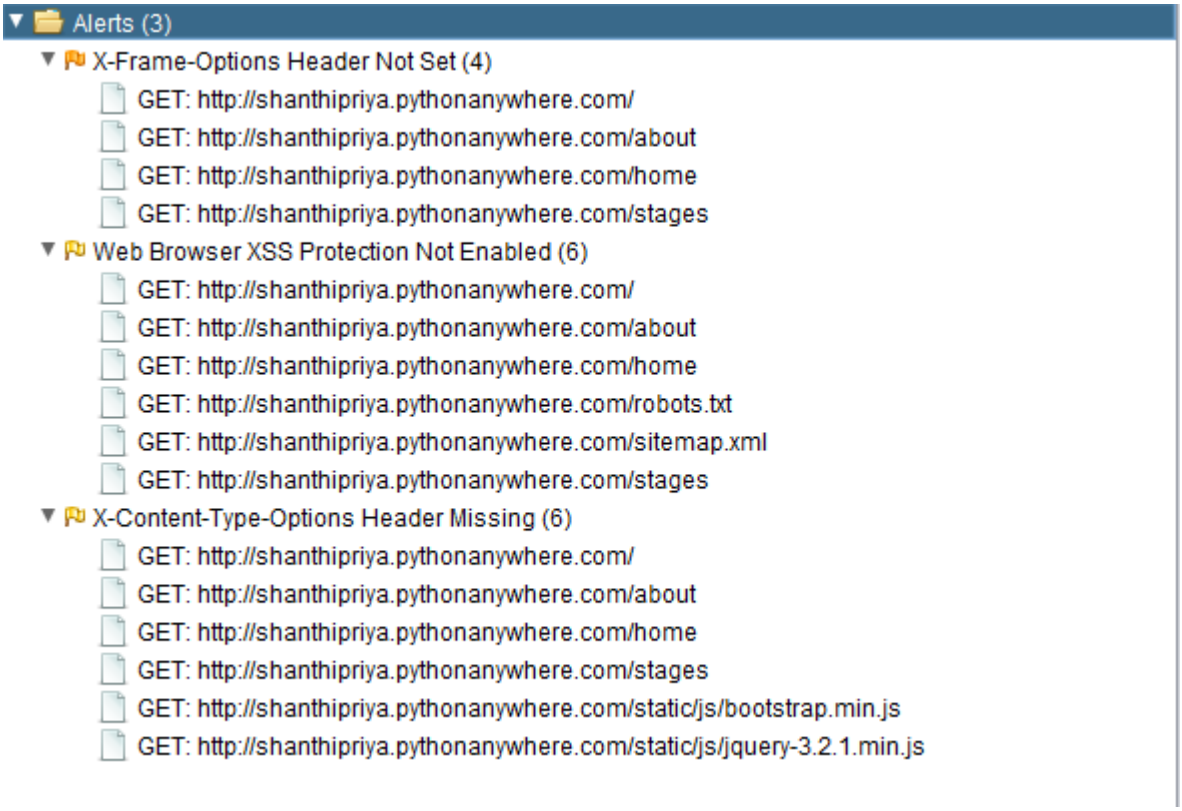


Figure 5. Zed Attack Proxy scan report

Result Summary

The result gives overall 3 alerts in which there is 1 medium alert and 2 low level vulnerabilities. The OWASP Zed Attack Proxy (ZAP) is one of the most popular free automated security tools, as per their claims it will give information regarding various common vulnerabilities, but according to the result it is not clear regarding the attack that have been performed in the website, as it does not give adequate information about Cross-site scripting(XSS) vulnerabilities. The result just gives some general description about the vulnerabilities that have been found during the test, one of the important information I have obtained from the scan is – "Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server". This brief result might not be adequate for junior level penetration testers.

B. Acunetix

Acunetix is a fully automated web vulnerability scanner that detects and reports on over 4500 web application vulnerabilities including all variants of SQL Injection and XSS. It fully supports HTML5, JavaScript and Single-page applications as well as CMS systems. It includes advanced manual tools for penetration testers and integrates with popular Issue Trackers and WAFs.

Result

Scan of shanthipriya.pythonanywhere.com

Scan details

Scan information	
Start time	19/02/2019, 19:47:29
Start url	http://shanthipriya.pythonanywhere.com/
Host	shanthipriya.pythonanywhere.com
Scan time	9 minutes, 9 seconds
Profile	Full Scan
Server information	openresty/1.9.15.1
Responsive	True
Server OS	Unknown

Threat level

Acunetix Threat Level 1

One or more low-severity type vulnerabilities have been discovered by the scanner.

Alerts distribution

Total alerts found	3
High	0
Medium	0
Low	2
Informational	1

Figure 6. Acunetix scan report-1

Web Server	
Alert group	**Clickjacking: X-Frame-Options header missing**
Severity	Low
Description	Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. The server didn't return an **X-Frame-Options** header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.
Recommendations	Configure your web server to include an X-Frame-Options header. Consult Web references for more information about the possible values for this header.
Alert variants	
Details	Not available in the free trial
Not available in the free trial	

Web Server	
Alert group	**OPTIONS method is enabled**
Severity	Low
Description	HTTP OPTIONS method is enabled on this web server. The OPTIONS method provides a list of the methods that are supported by the web server, it represents a request for information about the communication options available on the request/response chain identified by the Request-URI.
Recommendations	It's recommended to disable OPTIONS Method on the web server.
Alert variants	
Details	Not available in the free trial

Figure 7. Acunetix scan report-2

Result Summary

The result is closely similar to ZED proxy tool, however the alert level differs in acunetix. In most of the penetration testing tools the vulnerabilities are resolved based on the alert level. According to many available resources regarding penetration testing tools, in most cases the low level and informational level vulnerabilities are not taken into serious consideration.

In this case the vulnerability that has been classified in low level has been classified in medium alert level in Zed Proxy scan tool, so the vulnerability resolving issue might differ in both cases. By ignoring the low level vulnerability might cause serious vulnerability issues in the later days. Overall, acunetix free version tool failed to identify the direct vulnerabilities related to Cross-site scripting (XSS) and does not give detailed information for further understanding of the vulnerabilities that have been found.

C. Vega

Vega is a free open source web application testing tool. Written in JAVA, Vega comes with GUI interface. It is available for Windows, Linux, and Mac OS. It includes an automated scanner and an intercepting proxy. It can detect web application vulnerabilities as follows:

- Find SQL injection
- Validate SQL injection
- File inclusions
- Cross-Site Scripting (XSS)
- Improve security of TLS servers

It can be extended through a javascript API.

Result

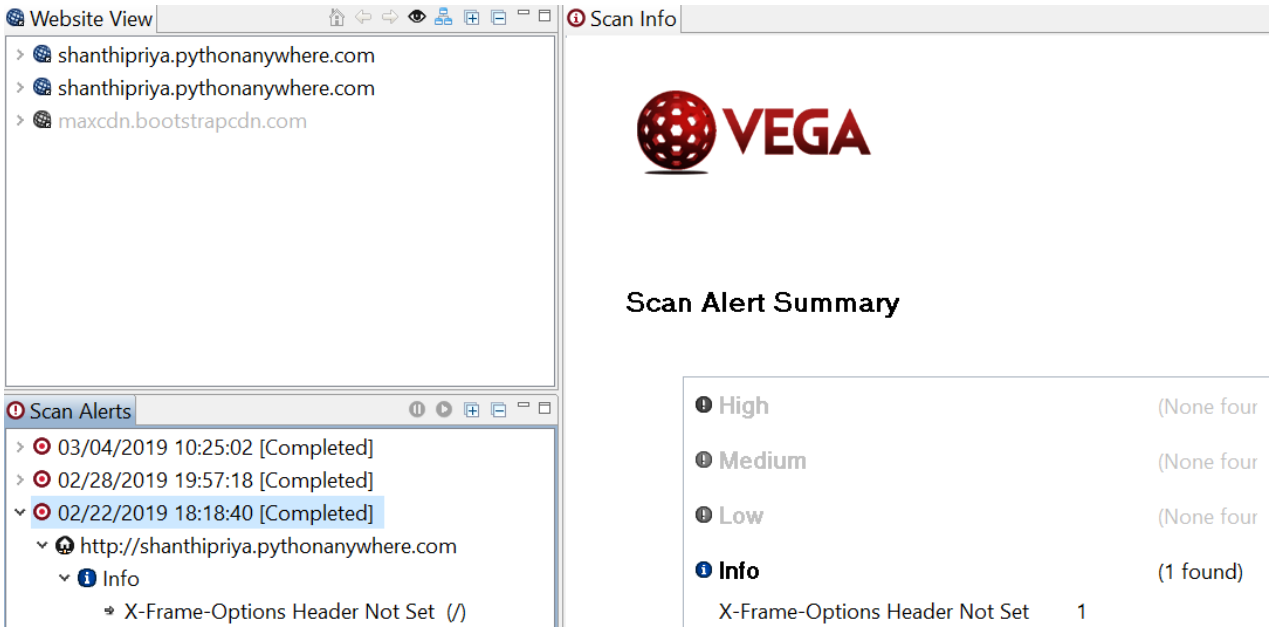


Figure 8. VEGA scan report-1

Result Summary

The Obtained result is similar in finding vulnerability, however the alert level is completely different from the previous tested tools. As mentioned earlier the alert level is a crucial information taken into account for resolving the issues. By letting the vulnerability unresolved it might be door for the hackers to cause critical issues.

Once resolving the XSS vulnerability in the website, it is again tested in the same tool for comparison and the obtained result is as follows:

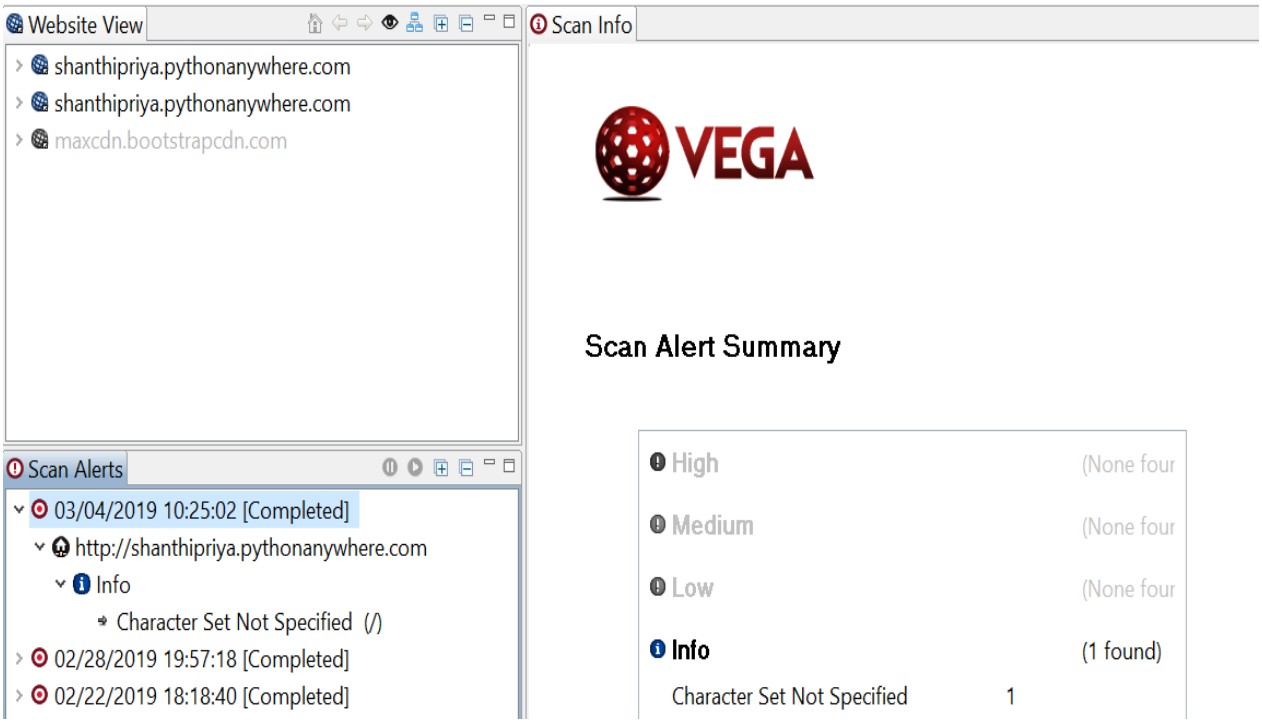


Figure 9. VEGA scan report-2

Result Summary

From the Alert summary it is clear that the vulnerability is resolved however, it has given different info alert which was not found in the earlier test, since there has been no changes except the solution of XSS attack, and the conclusion is that the tool failed to detect all the issues in the first scan.

7.3 Testing Online Tools

Tools that have been discussed in the previous sections are some of the installed free open source web application security testing tools.

In this section some of the online penetration testing tools will be tested and compared.

A. Pentest-Tools.com

Pentest-Tools.com is an online penetration testing tool which was started in 2013. This tools checks for various web application vulnerabilities which includes:

- SQLiScan
- XSS Scan
- URL Fuzzer
- Web Server Scan
- CMS Tests

Since I have performed XSS attack I am using XSS scan tool and the result is as follows:

Result

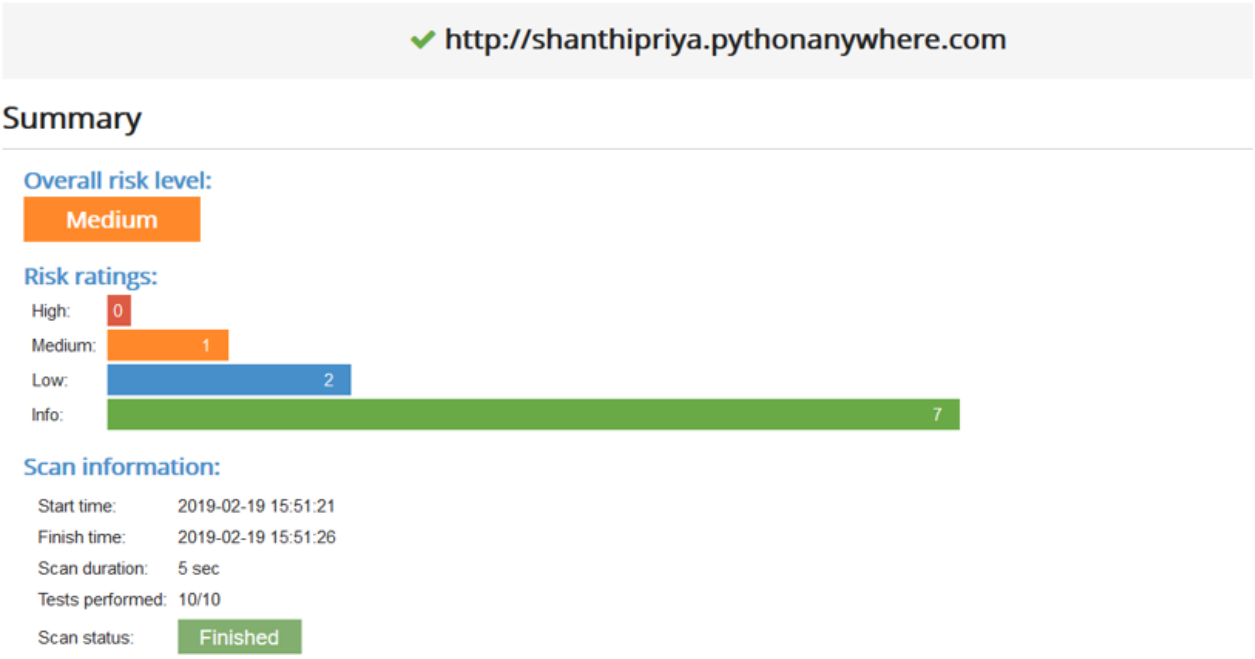


Figure 10. Pentest-Tools.com scan report-1

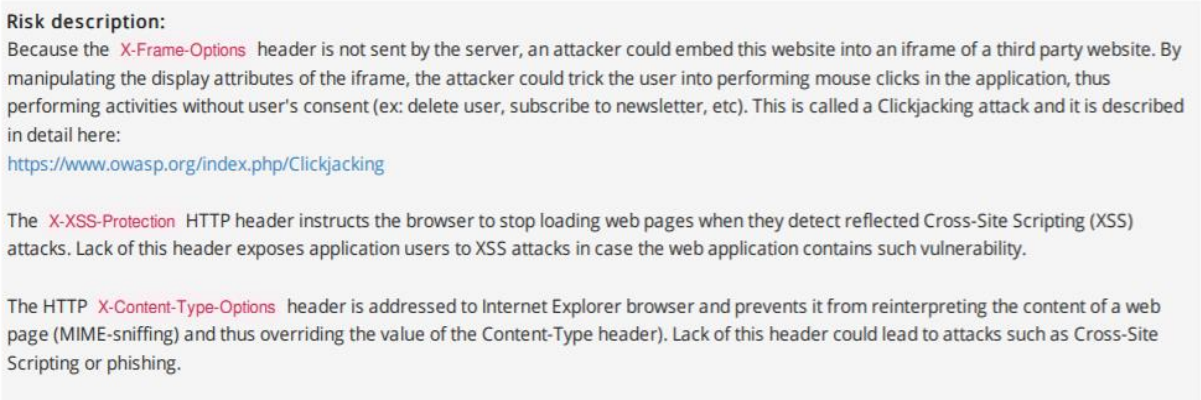


Figure 11. Pentest-Tools.com scan report-2

Result Summary

The result summary is similar to Zed proxy scan tool, as there are 1 medium level, 2 low level and 7 info level vulnerabilities found. This is the first tool that has given detailed information regarding issues in the website compared to the previous tested tools, and as we discussed in the previous section the alert level information are really important during penetration testing and this tool was able to discover the maximum vulnerabilities in the website and their respective vulnerable level. With reference to Figure. 11 the risk description gives detailed information regarding how this vulnerabilities that have been found pave way for

mitigating cross-site scripting (XSS) attacks in the website.

The following result gives information after resolving the vulnerabilities in the website. As I am using light (free) version the only way to make sure that the cross-site scripting (XSS) related issues have been resolved is by obtaining “no cross-site scripting (XSS) attacks found” message in the findings section.

Result

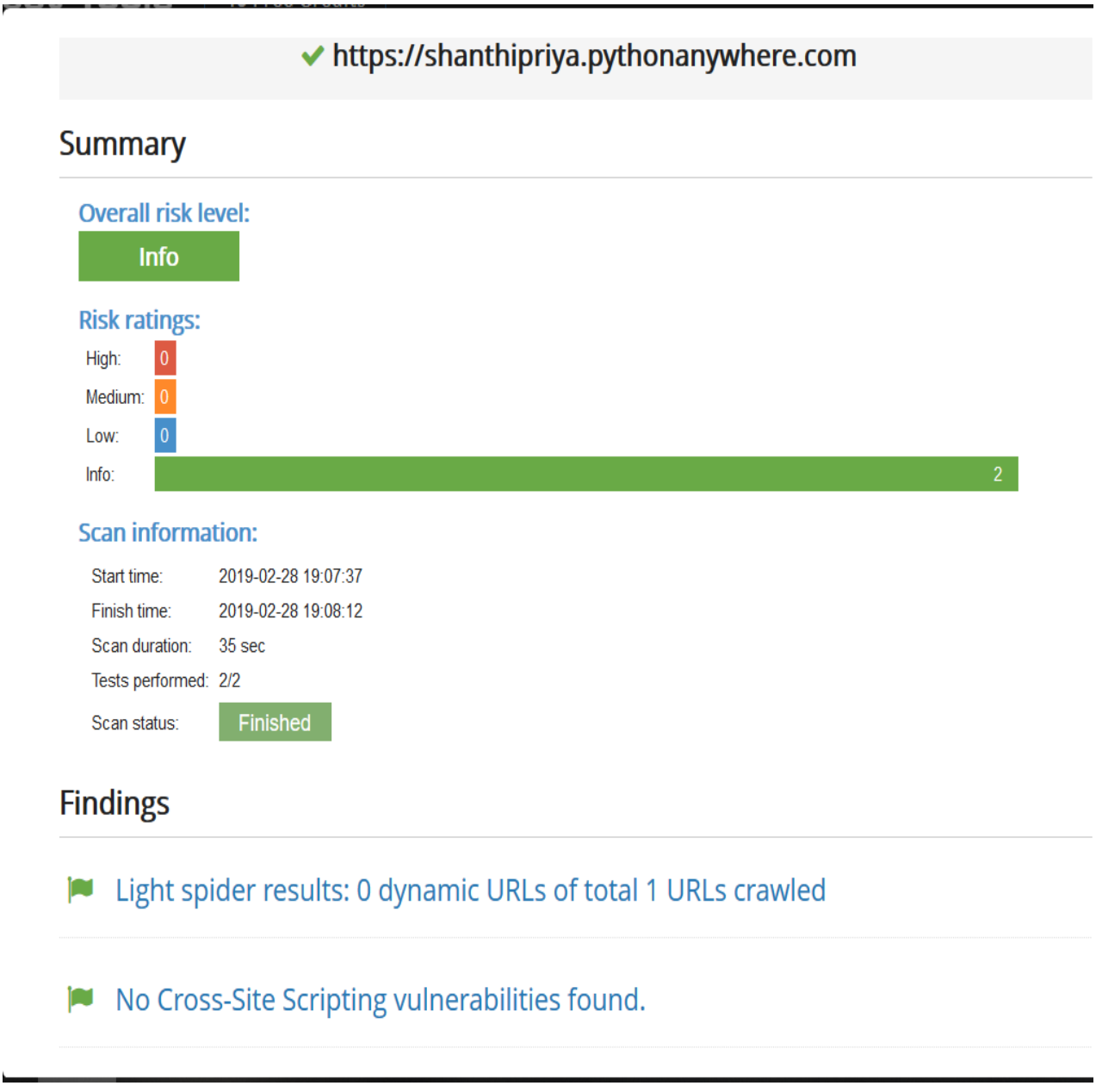


Figure 12. Pentest-Tools.com scan report-3

Result Summary

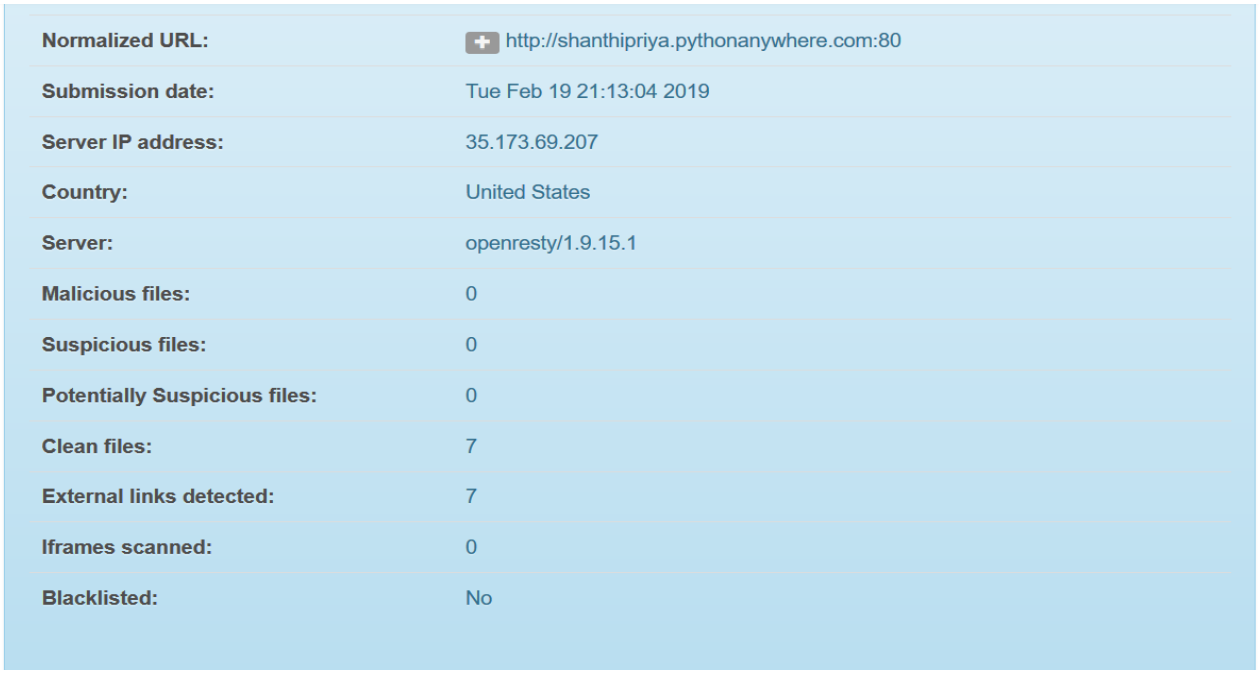
With respective to the result it is clear that the issues have been resolved, and in the finding section it is clear that no cross-site scripting vulnerabilities are found. This online tools also gives various options for various vulnerability scans and might be comfortable even for less experienced testers as it gives adequate and clear information's regarding vulnerabilities.

B. Quttera

The Quttera Web Malware Scanner plugin will scan the website for malware, trojans, backdoors, worms, viruses, shells, spyware and other threats as well as JavaScript code obfuscation, exploits, malicious iframes, malicious code injection, malicious code obfuscation, auto-generated malicious content, redirects, hidden eval code and more. Also, it will check whether the website is blacklisted by Google and other blacklisting authorities. Helps to protect the website, the website users and online reputation with a free Quttera Web Malware Scanner plugin.

Features:

- One Click Scan
- Unknown Malware Detection
- External Links Detection
- Blacklist Status
- No Signatures or Patterns Updates
- Artificial Intelligence Scan Engine
- Cloud Technology
- Detailed Investigation Report
- Investigation of WordPress files
- Detection of files infected by PHP malware
- Detection of injected PHP shells



Normalized URL:	http://shanthipriya.pythonanywhere.com:80
Submission date:	Tue Feb 19 21:13:04 2019
Server IP address:	35.173.69.207
Country:	United States
Server:	openresty/1.9.15.1
Malicious files:	0
Suspicious files:	0
Potentially Suspicious files:	0
Clean files:	7
External links detected:	7
Iframes scanned:	0
Blacklisted:	No

Figure 13. Quttera scan report

Result Summary

The above result has completely ignored all the misconfigurations and no vulnerabilities have been found, Except for the pentest Tool.com most of the online tool I have researched are not giving detailed information and ignoring maximum vulnerabilities.

8. COMPARISON OF EXISTING TOOLS

From the penetration test among various tools there are some common vulnerabilities that are being tested based on the claims in the tool's website. According to the results obtained from various tools there are some variations in the results, main difference is in the level of risk of the vulnerabilities that have been found. Based on the detailed references with various available online tools most of them are not generating detailed test results, and only brief information are obtained. Quttera scan online tool is one the example for such issues.

With reference to the experiment results and available researches, regarding penetration testing, most of the tools are performing some common tests which includes:

- SQLScan
- XSS Scan
- CSRF
- Buffer Flow
- External Links Detection
- Blacklist Status
- File inclusions

Even though they claim to test for these vulnerabilities the results from the tools are not detailed. For example according to Zed Attack Proxy they perform Automatic Scanner Authentication support, AJAX spiders, Dynamic SSL certificates, Forced Browsing, Intercepting Proxy, Web Socket Support, Plug-n-hack support, REST-based API and much more. But based on the report it is clear that most of the tests performed were related to XSS and HTTP configuration. The Acunetix tool gave brief theoretical explanation of the importance of port scanning but, from the obtained result there is no such information regarding network configurations.

Based on the conducted experiment on XSS attacked website, the comparison of results are as shown in Table 6.

Tools	XSS	Alert Level	Detailed Information
Zed Attack Proxy	Yes	Medium - 1, Low - 2	Yes
Acunetix	Yes	Low – 2, Info-1	Yes
Vega	Yes	Info - 1	No
Pentest-Tools.com(Online Tool)	Yes	Medium – 1 Low – 2 Info - 7	Yes
Quttera	No	-	No

Table 6. Result comparison of XSS affected website in various Tools

As we know network configuration in one of the important factor that has to been taken into consideration during checking for vulnerabilities or gateway that may be vulnerable for attacks most of these tools are not giving any detailed information about port status and configurations. One of the famous port scanner is Nmap which gives detailed information about the ports and gives their status and detailed risk rating if its safe to be open or not. But there are not many web application scanner which scan for these details mentioned above. So, I have implemented a web application port scanner which scans and reports the status of the ports from which the penetration tester can be aware of open ports and check if it's safe to be open or not.

8.1 Importance of Port checking

- If unintended services for instance debug mode is running on a server and has been identified during the port scan, the attacker can find them and try to connect to them.
- If intended services are running on a server but not on their standard ports (i.e. ssh running on 2222/tcp instead of 22/tcp), a "security through obscurity" play and that can increase the attack surface
- There are some situations where port scanning using different techniques can help reveal firewalls and other network controllers (by sending crafted packets and watching back for the results on open or closed ports etc.) that stands between the user and the server
- There are some situations where port scanning can be combined with banner grabbing and as such user can learn more about their target server (e.g. service software name, version, OS version etc.)

8.2 Benefits of TCP Scanning

Often hackers scan for open ports to see which open ports they may want to exploit. Once a hacker finds an open port, they often use particular hacker programs that are uniquely coded to exploit a particular port. If a port is open, the operating system completes the TCP three-way handshake, and the port scanner immediately closes the connection to avoid performing a Denial-of-service attack. Otherwise an error code is returned.

When a request is made to an open port on the perimeter server, the computer sends a response back letting the requester know that there is a service listening on this port. So what this means is that anything that comes through this port will be processed by a service. When a request is made to a closed port, the server replies that no services are listening on that port. Finally when the request is made, no reply is sent at all and all packets are dropped then breaking into one's network is very hard.

So, as discussed in chapter 2, I have implemented an AppScanner tool for Web Application in which the hidden directories will be checked. Which will be helpful to obtain more information about the site and it might also be used for tracking of user information, because hidden directories and files left accidentally on the web server might be a very valuable source of sensitive information. There can be a lot of hidden information in web application root folder which are, source code version systems folders, files (.git, .gitignore, .svn), project configuration files (.npmrc, package.json, .htaccess), custom configuration files with common extensions like config.json, config.yml, config.xml and many others. Another function of this tool will be trying to find SQL vulnerability by injecting SQL payload finally it will be checking for open ports, by finding the ports status it would be really helpful to understand the service information running on the server. Hence the penetration tester can check if it's vulnerable or safe if the port is open.

9. IMPLEMENTATION

This chapters includes the practical implementation of the thesis, which includes the following tasks:

1. Implementing the Password protection tool, which will be safeguard tool for user login. Since privacy protection is a primary concern of GDPR, based on various conditions this tool will let the user to create a strong password.

2. Implementing Appscanner Tool, in which the tool will be scanning the website for any weakness in the website, the features of the Appscanner Tool are as follows:

   - Scans for Website IP, checks hidden directories and cloudflare error.
   - Perform SQL injection and checks for SQL injection vulnerabilities.
   - As discussed in chapter 8.1 and 8.2 knowing the port status is an important factor to make sure network is secure. Next process will be checking for the status of the port and saving all the gathered information in a text file as a report.

9.1 Implementation of Password Protection Tool

As discussed in chapter 4.4 regarding the importance of GDPR and there rules, I am trying to implement one of the regulation stated under regulation (EU) 2016/679 - "a high level of protection of personal data" is required. Even though passwords are not specifically mentioned, Passwords are an important safeguard to prevent unauthorized individuals from gaining access to sensitive data, so while not explicitly stated in the GDPR text, if passwords are used, they need to be covered by the GDPR policies.
GDPR is primarily concerned with improving privacy protections for EU citizens, which is achieved by ensuring any business, organization, or individual that handles the data of EU citizen's implements safeguards to preserve the confidentiality of that information. The purpose of a password is to prevent unauthorized individuals from accessing data or resources. Passwords can be considered an appropriate safeguard to ensure the security of accounts and the confidentiality of sensitive information, provided an appropriate GDPR password policy is in place. A weak password can easily be guessed and will be susceptible to brute force attacks. Strong passwords are therefore required. It is a good best practice to require passwords to be of a certain length, include upper and lower-case letters, a number, and a special character. A GDPR password policy should require passwords to be reset periodically. While not stated in the GDPR text, these measures to make passwords more secure are appropriate to the level of risk.

There are various guidelines for creating secure and strong passwords, from which I have chosen National Institute of Standards and Technology Special Publication 800-63B (NIST) Special Publication 800-63B guidelines.

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3551 et seq., Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

The guideline includes various points such as:

- 8 character minimum when a human sets it
- 6 character minimum when set by a system/service
- Support at least 64 characters maximum length
- All ASCII characters(including space) should be supported
- Truncation of the secret(password) shall not be performed when processed
- Check chosen password with known password dictionaries
- Allow at least 10 password attempts before lockout
- No complexity requirements
- No password expiration period
- No password hints
- No knowledge-based authentication (e.g. who was your best friend in high school?)
- No SMS for 2FA (use a one-time password from an app like Google Authenticator)

From the above mentioned guidelines I have chosen the following rules:

- 8 character minimum when a human sets it
- Support at least 64 characters maximum length
- Should contain minimum 1 Uppercase alphabetic character
- Should contain minimum 1 Lowercase alphabetic character
- Should contain minimum 1 Special character/Symbols
- Should contain minimum 1 Numeric value
- Check chosen password with known password dictionaries
- No complexity requirements
- No password expiration period
- No password hints

I will be concentrating on these points for my implementation as the remaining rules should be taken care by the database administrator.

Structure of tool is as shown in Figure 13.

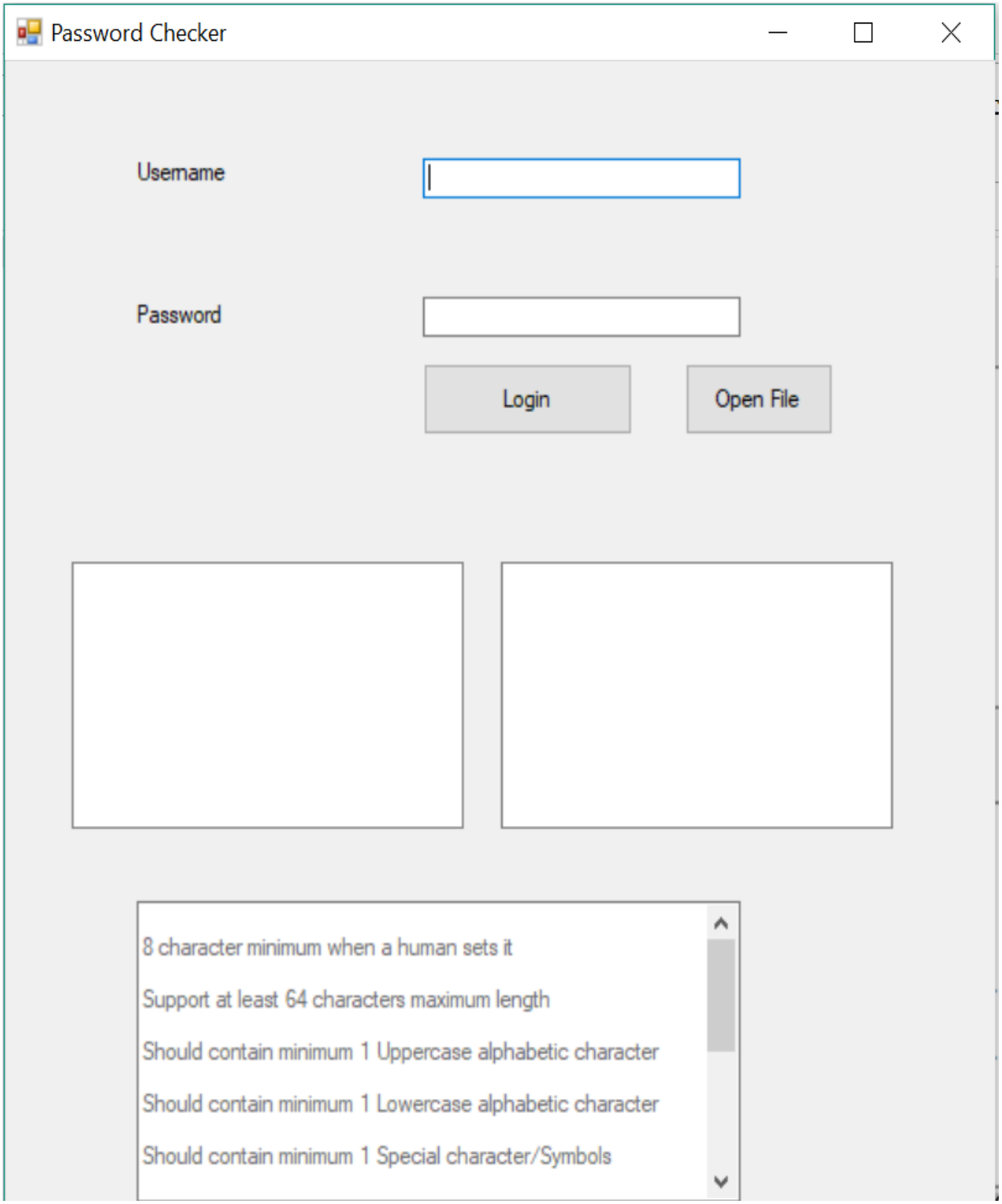


Figure 14. Password Protection Tool

In My implementation, I have created various conditions for password with reference to NIST guidelines as mentioned in chapter 9.1, which includes basic conditions like containing upper case, lower case, symbols, numeric values, password should not contain 3 consecutive letters as username, the user can upload common or frequently used password list file using open file button and it will be uploaded to the local database and finally while login function is performed the password checker tool will check if the conditions are satisfied and the results will be shown in green color if they are satisfied, if not the results will be displayed in red color as shown in Figure 15.

Result

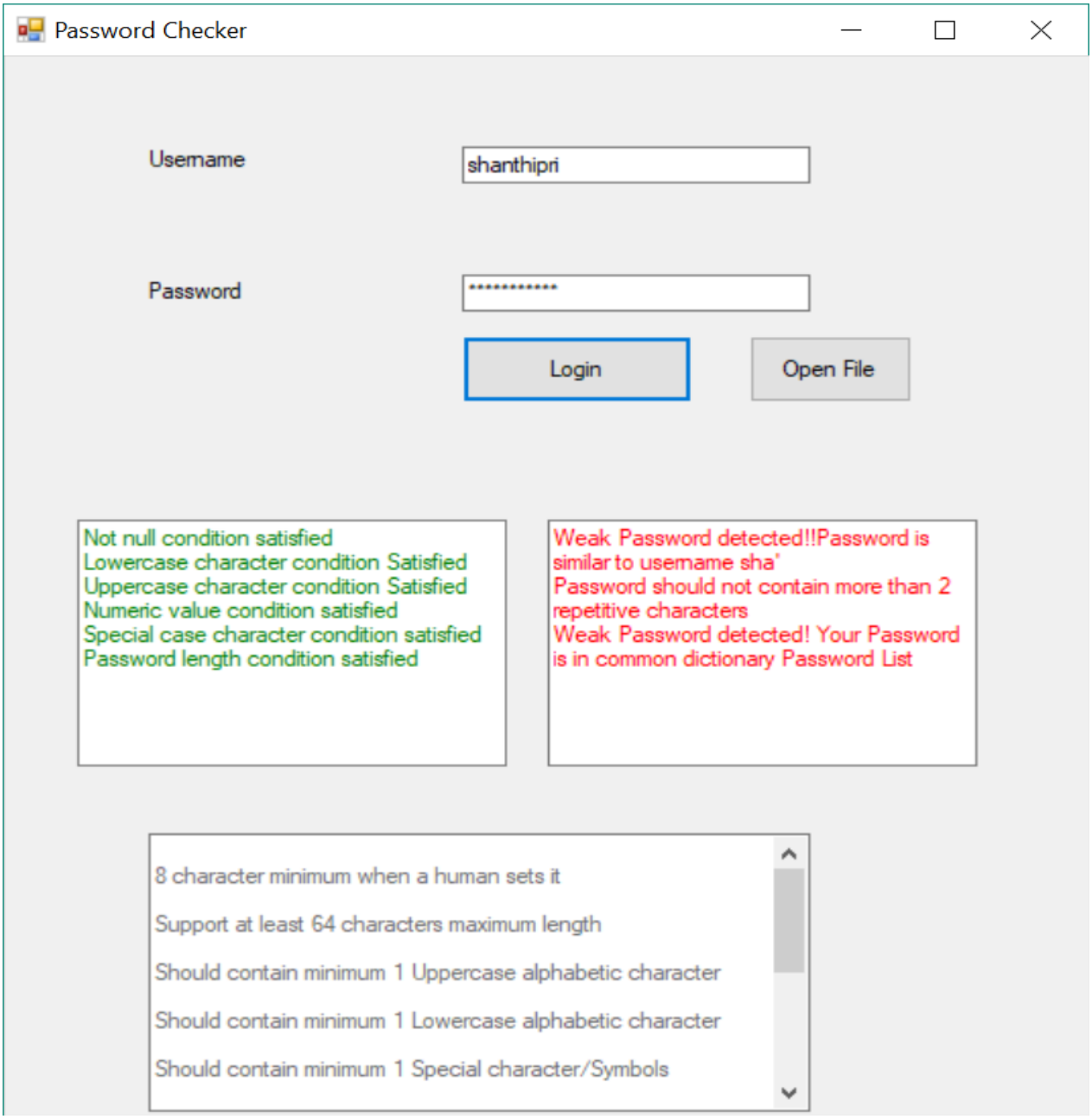


Figure 15. Result for Password Protection Tool

9.2 Implementation of AppScanner Tool

As discussed in chapter 8.2 and 8.3 the implementation of AppScanner tool is as follows

Structure of tool is as shown in Figure 16.

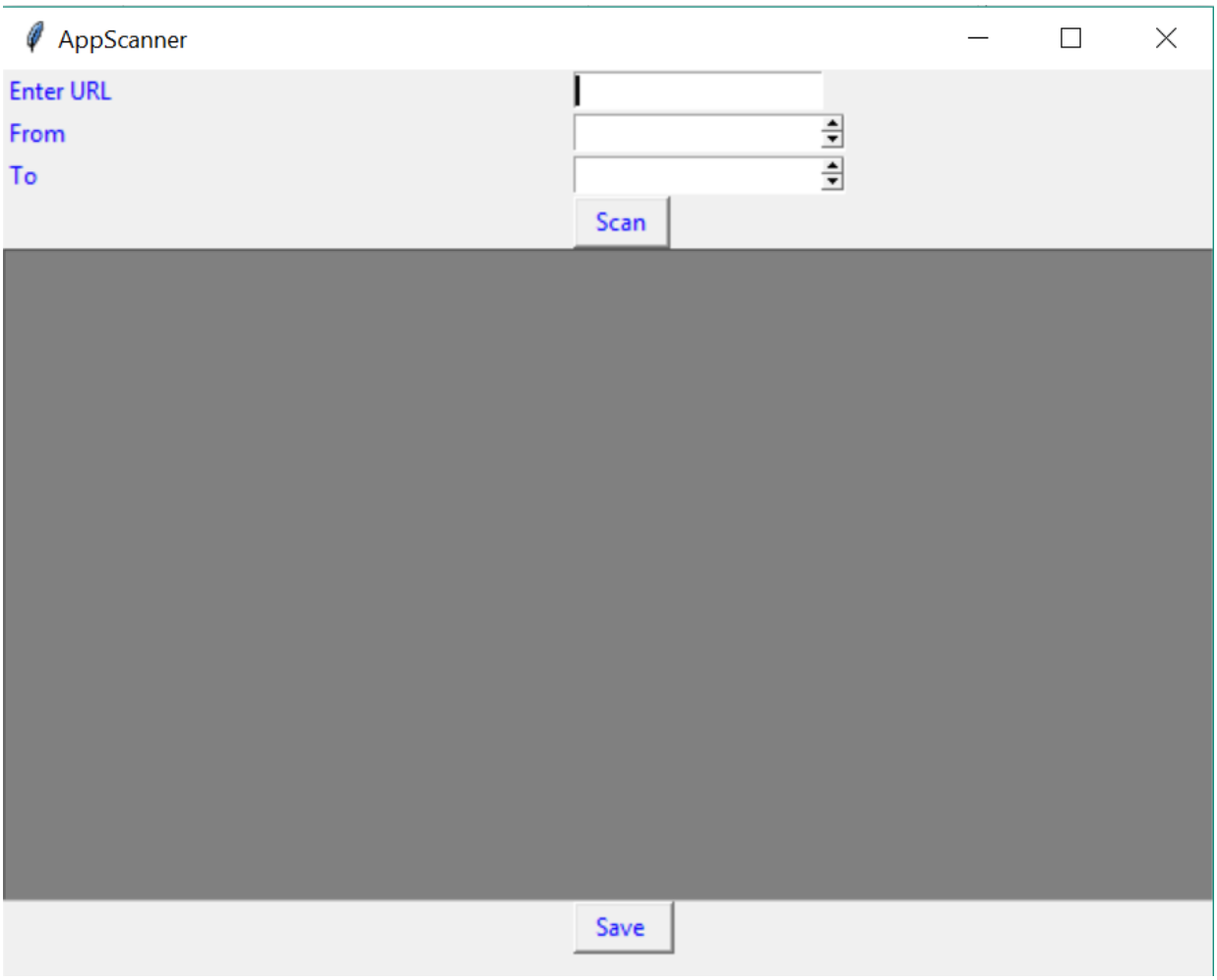


Figure 16.AppScanner Tool

This tool is designed for ports checking of the given website or IP address. Tool checks for hidden directories, and perform SQL injection to find SQL vulnerability in the website and finally the gathered information (report) will be stored as a file.

Result

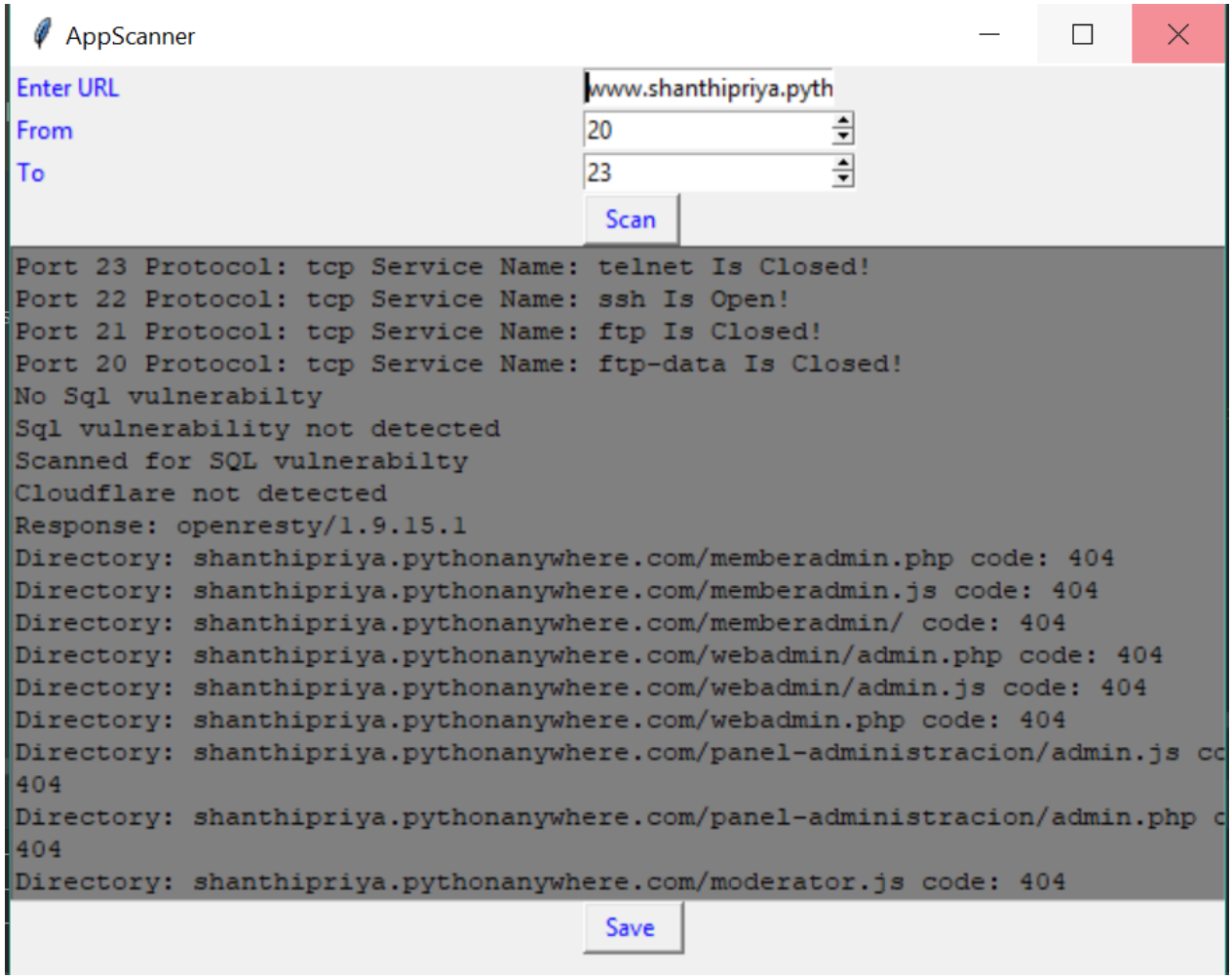


Figure 17. AppScanner Report

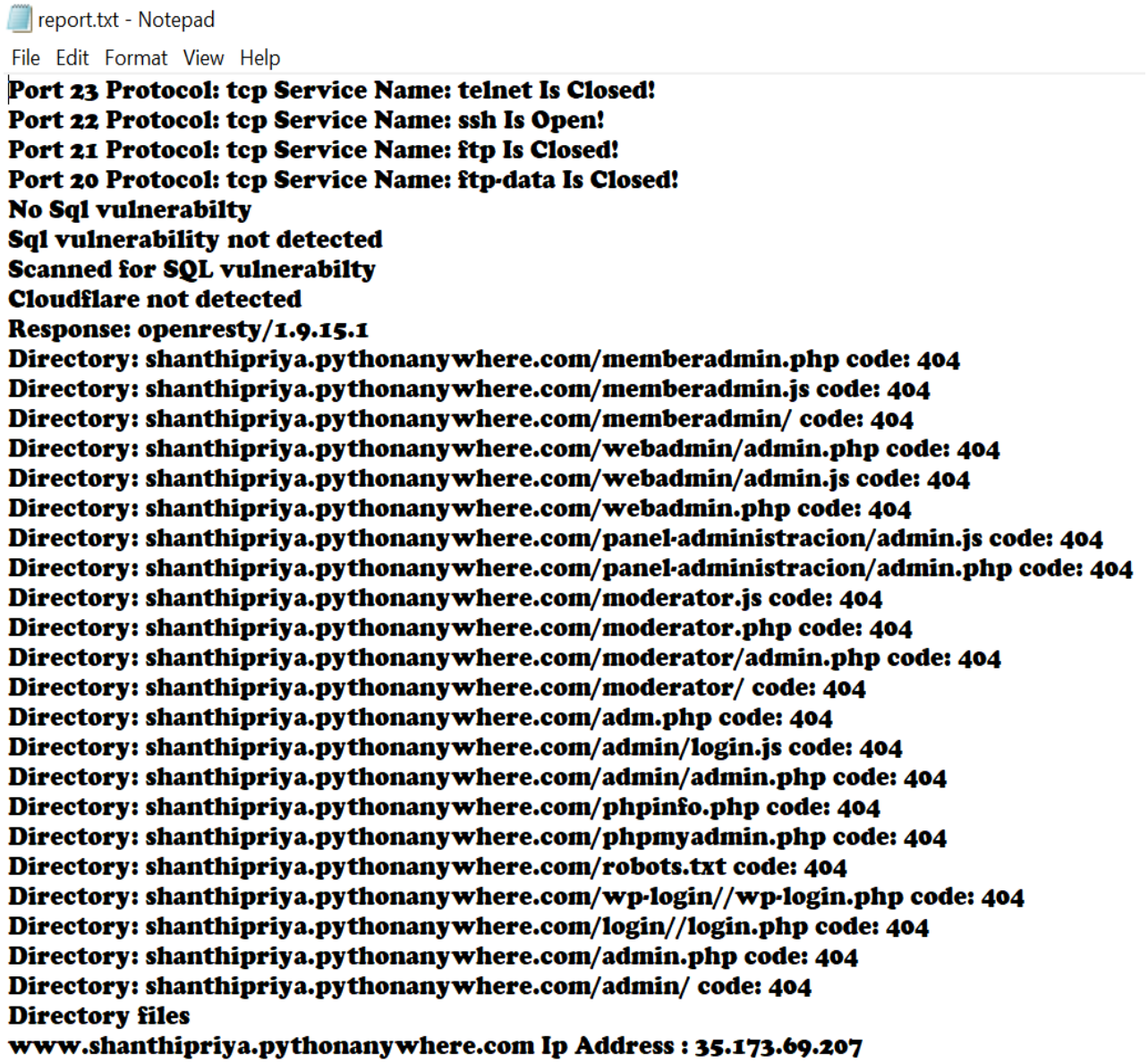


Figure 18. Generated and stored final- scanning report

I have scanned the ports from port 20 to port 23 and as we can see Port 22 is in open status and the remaining ports are closed. Port 22 is a Secure Shell port most commonly used in command line access, secure replacement of Telnet. Could also be used as an encrypted tunnel for secure communication of virtually any service [RFC 4251].

The Secure Shell (SSH) Protocol is a protocol for secure remote login and other secure network services over an insecure network. The SSH protocol consists of three major components: The Transport Layer Protocol provides server authentication, confidentiality, and integrity with perfect forward secrecy. The User Authentication Protocol authenticates the client to the server. The Connection Protocol multiplexes the encrypted tunnel into several logical channels. The Basic TCP port mechanism will be helpful to understand the mechanism clearly and explains how the ports are identified as open and closed.
The following chapter explains the TCP port mechanism in detail.

9.3 TCP Port Mechanism

This mechanism essentially involves the exchange of three packets, as follows:

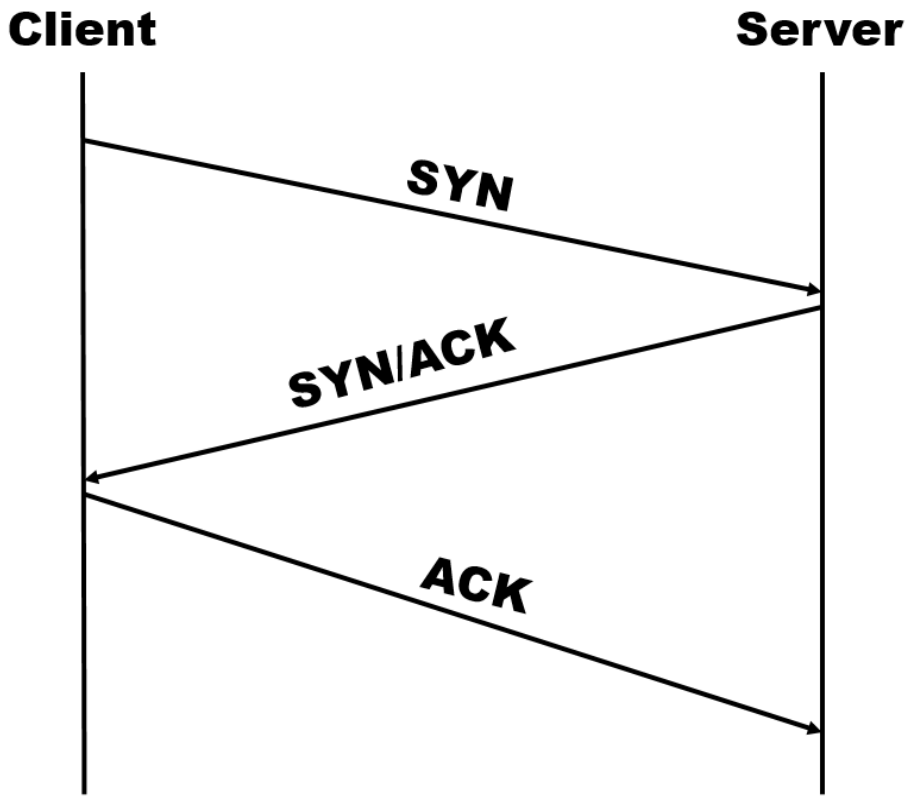


Figure 19. Packet exchange mechanism

- The client sends a SYN packet to the server. This is a TCP packet with the SYN flag set, and it contains important information, such as the initial sequence number and desired TCP options.
- The server responds with a SYN/ACK packet, namely a TCP packet with both the SYN and ACK flags set. This packet includes information similar to the client's SYN packet and acknowledges receipt of the client's SYN.
- The client responds with an ACK packet that acknowledges the receipt of the SYN/ACK from the server.

9.3.1 Mechanism of exchanging Packets if the port is closed

If the target port happens to be closed, the server will instead respond with a reset (RST) to the client's SYN, as seen in Figure 18.

Thus, one can tell the state of a TCP port by sending a SYN packet to the target system and port, and then wait for the response packet; a SYN/ACK response indicates the port is open, while an RST indicates the port is closed.

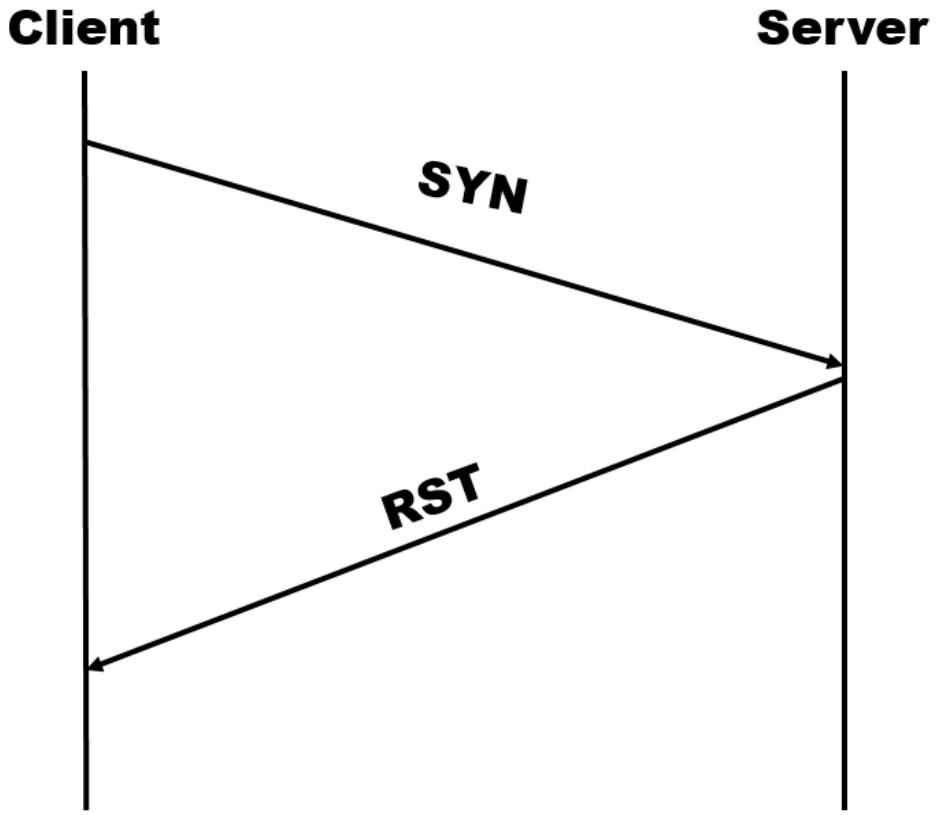


Figure 20. Packet exchange mechanism while port closed

9.3.2 Forging SYN packets as part of port scan attacks

If attacker employed a forged IP source address for his SYN packet. If the port was open, the target server would respond with a SYN/ACK, as expected. However, this SYN/ACK would be directed to the forged source address of the SYN packet, rather than to the attacker's IP address.

Upon receipt of the SYN/ACK, the receiving system would find this packet bogus as it never actually sent a SYN packet and respond with an RST.

If the target port at the server was closed, the server would respond with an RST, as expected. As in the previous scenario, this RST packet would be directed to the forged IP address, rather than to the attacker.

However, TCP packet-processing rules indicate that RST packets do not elicit any packets. If they are deemed bogus, however, they are silently dropped. This scenario is illustrated in Figure 19.

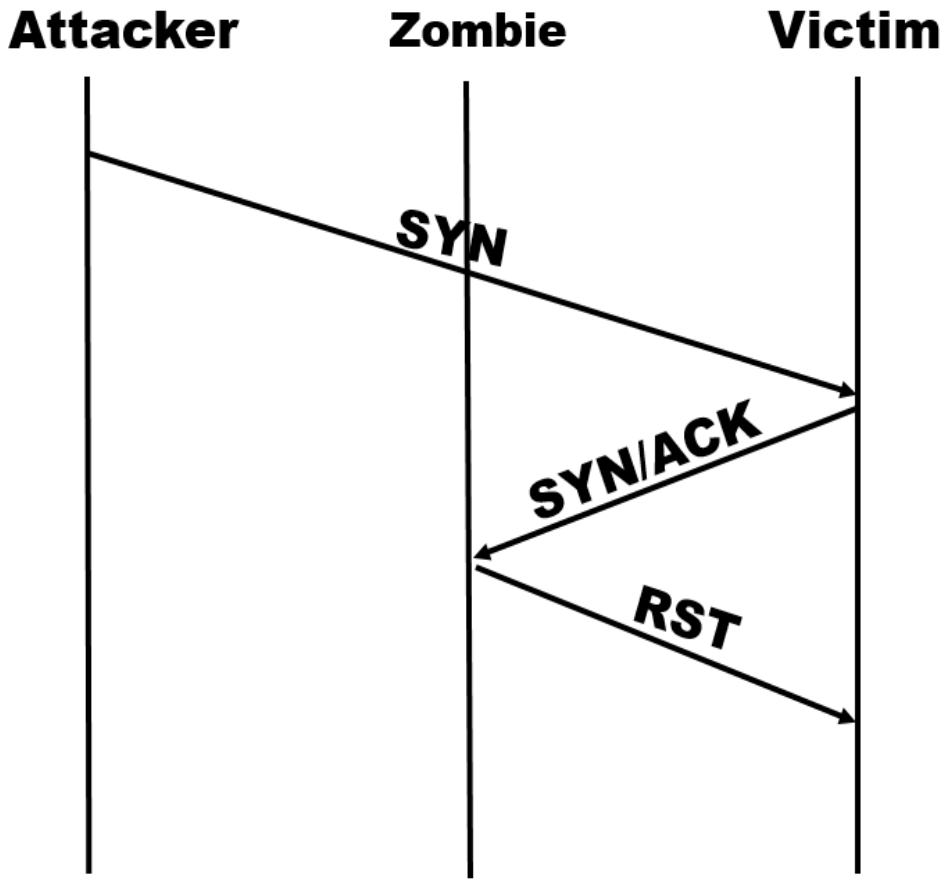


Figure 21. Forging SYN packets as part of port scan attacks

If the attacker was able to tell whether his probe packet -- the forged SYN packet -- caused the zombie to send a packet, he could determine if the port was open or closed. Put another way, the response from the server could reveal whether the zombie packet was sent as a result of the forged SYN packet.

9.3.3 Information leakage at the IP layer

The key mechanisms of the Internet Protocol are those of fragmentation and reassembly: They allow the Internet Protocol to operate over a wide range of technologies by carving up an original packet into smaller packets, or fragments, and reassembling those pieces into the original packet at the destination system.

The Identification field of the IP header labels packet fragments. The values in that field such as IP source address, IP destination address, protocol and identification are then used by the receiving system to reassemble the fragments into the original packet.

Hosts typically use a different identification value for each transmitted packet. More specifically, they usually set the Identification field to a global counter that tracks each transmitted packet. Each packet is counted, including those not fragmented by the source system.

As a result, a system receiving a flow of packets from another system could see an incremental sequence of IP Identification values, such as 44567, 44568, 44569 and 44570. If, at any point of that sequence, some

value was skipped, that would very likely be an indication that the sending host sent one packet to some other system. Put another way, IP implementations that employ a global counter for setting the IP Identification field essentially leak information about the number of packets they transmit.

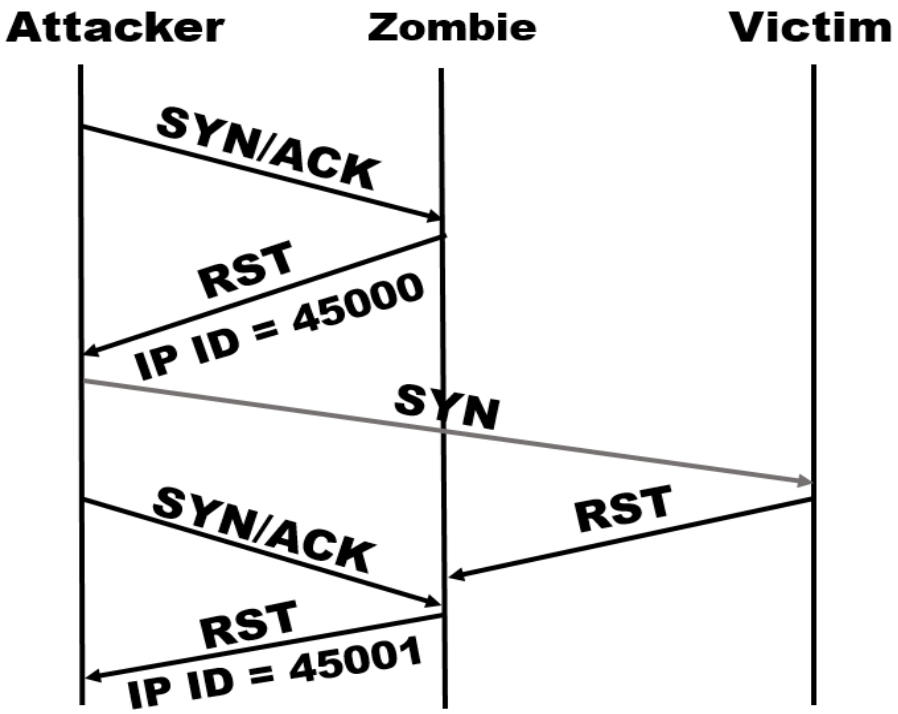


Figure 22. Consecutive values indicate the TCP port is closed.

9.3.4 Idle scan and how it can be exploited

Anonymous TCP port-scanning techniques can be implemented by putting together all the concepts discussed above. This technique can be thought of having three actors: the attacker, the victim and the zombie. The attacker is the host performing the port scan attacks, while the victim will be the target. Finally, the zombie will be a system exploited by the attacker to conceal the port scan.

It works like this: The attacker will sample the current IP Identification value at the zombie. Then, he'll send a forged SYN packet. He'll then sample the IP Identification again.

Such sampling can be performed by sending any packet that would elicit a response from the zombie, such as a SYN/ACK. If the sampled IP Identification values were consecutive -- like 45000 and 45001 -- that would be an indication the zombie did not send any packets as a result of the forged SYN packet and, thus, the corresponding TCP port was closed. The packet exchange corresponding to this scenario is illustrated in Figure 20.

On the other hand, if the corresponding IP Identification values were not consecutive -- say, 45000 and 45002 -- this would indicate the transmission of one packet by the zombie node. This would also mean the corresponding port was open. Figure 21 illustrates the corresponding packet exchange.

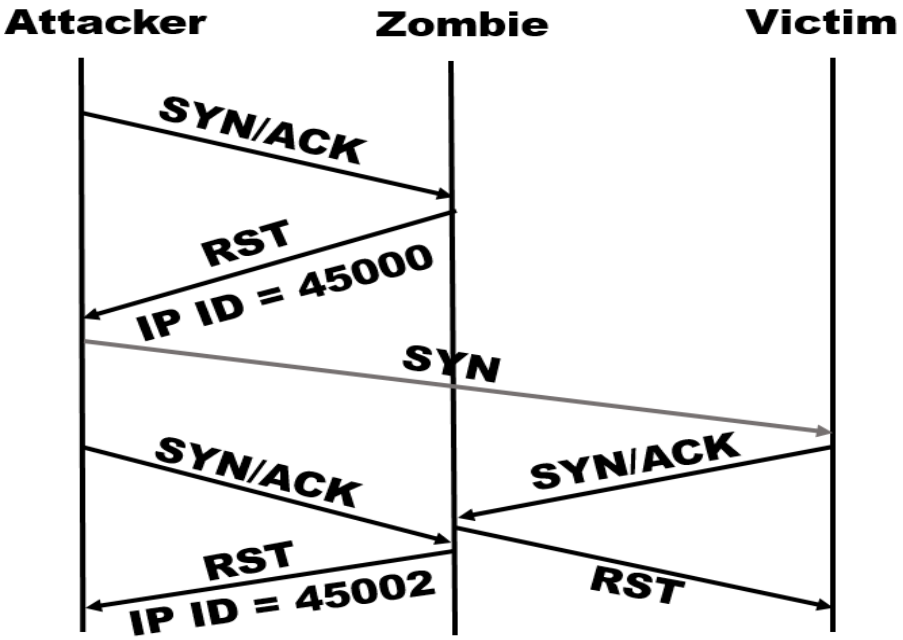


Figure 23. Nonconsecutive values indicate the TCP port is open.

There are some different scanners with different ranges of functions and the most of these tools are available as freeware or open source options. Many of these feature classic command-line programs, so I have implemented a GUI Port Scanner Tool through which the tester can understand the status of the port. Ports are one of the main reasons that data packets are able to find their way to their desired destination. They serve as interfaces between computers and system services or programs and are used by the network protocols. Together with the IP address, operating systems are able to find out which computer and application they're meant to send the data. Now, Let us take a look into some more common Open ports and their uses in Table 7.

9.4 Common Open Ports

Some of the common Ports no, name and their brief description are as follows

Port No	Port Name	Description
TCP 20 and 21	File Transfer Protocol, FTP	The FTP protocol uses a pair of connections between the FTP client and FTP server. The connection with the FTP server's port 20 is the second connection created during an FTP session, the first one being to the server's port 21. File Transfer Protocol (FTP) is one of the oldest Internet protocols. FTP servers open their machine's port 21 and listen for incoming client connections. FTP clients connect to port 21 of remote FTP servers to initiate file transfer operations.
TCP 22	Secure Shell, SSH	An Encrypted replacement for Telnet (and in some

		cases ,FTP)
TCP 23	Telnet	Telnet is one of the earliest, original protocols of the Internet. A machine offering Telnet services is essentially offering to accept an "across the Internet" remote console terminal connection from any client device. This makes Telnet quite powerful and, without proper security, a significant security concern.
TCP 25	Simple Mail Transfer Protocol, SMTP	SMTP is the protocol used to shuttle eMail across the Internet from one mail server to another. Over its years of use, the protocol has evolved significantly to become much more capable, and much less "simple" than it was in the beginning. SMTP servers open and listen for incoming connections on port 25. Another SMTP server, or a personal eMail client, will connect to the server on its port 25 to transfer some eMail into it for subsequent forwarding toward its destination.
TCP and UDP 53	Domain Name System, DNS	It is used for domain transfers and DNS servers also listen on UDP port 53 to accept queries from client resolvers.
UDP 69	Trivial File Transfer Protocol, tftp	Trivial File Transfer Protocol - A less secure version of FTP, generally used in maintaining and updating systems, for configuration file transfers between LAN systems, firmware updates on routers, etc.
TCP 79	finger	User Information Protocol- Finger servers provide information about the users of their computers by opening and listening for incoming TCP connections on port 79. Remote users wishing to obtain information about the user of a specific computer could do so by querying their machine's finger server listening on port 79. This information typically included the user's full name, address, telephone number, title, job name, office location, telephone extension, and so on.
TCP 80	Hypertext Transfer Protocol, HTTP	This is the primary port used by the world wide web (www) system. Web servers open this port then listen for incoming connections from web browsers.

		Similarly, when a web browser is given a remote address (like grc.com or amazon.com), it assumes that a remote web server will be listening for connections on port 80 at that location.
TCP 110	Post Office Protocol v3, POP3	Pop3 "post office protocol" is used by eMail clients for the retrieval of their eMail from designated eMail "post office" servers. Email Clients such as Microsoft Outlook, Netscape, Eudora, and many others, connect to port 110 of a remote eMail server, then use the pop3 protocol to retrieve their eMail. They first identify and authenticate themselves by logging on to the remote eMail server using their eMail account information. After doing so they are permitted to view and download their waiting eMail.
TCP 119	Network News Protocol, NNTP	Port 119 hosts the servers of the famous and infamous Internet USENET newsgroup world. NNTP servers push and pull news articles to and from other NNTP servers over port 119, and news reading (and writing) clients talking to news servers over the same port.
UDP 161 and 162	Simple Network Management Protocol, SNMP	Simple Network Management Protocol (SNMP) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior. Devices that typically support SNMP include cable modems, routers, switches, servers, workstations, printers, etc.
UDP 443	Secure Sockets Layer over HTTP, https	The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access.

Table 7. Common Open Ports

9.5 Some of the commonly affected Bad/Vulnerable ports

Port No	Backdoors or Hacker's remote access tools
TCP 31	Agent 31, Hackers Paradise, Masters Paradise
TCP 1170	Psyber Stream
TCP 1234	Ultors Trojan
TCP 1243	SubSeven server (default for V1.0-2.0)
TCP 1981	ShockRave
TCP 2001	Trojan Cow
TCP 2023	Ripper Pro
UDP 2140	Deep Throat, Invasor
TCP 2989	Rat backdoor
TCP 3024	WinCrash
TCP 3150	Deep Throat, Invasor
TCP 3700	Portal of Doom
TCP 4950	ICQ Trojan
TCP 6346	Gnutella
TCP 6400	The Thing
TCP 6667	Trinity intruder-to-master and master-to-daemon SubSeven server (default for V2.1 Icqfix and beyond)
TCP 6670	Deep Throat
TCP 12345	NetBus 1.x, GabanBus, Pie Bill Gates, X-Bill
TCP 12346	NetBus 1.x
TCP 16660	Stacheldraht intruder-to-master
UDP 18753	Shaft master-to-daemon
TCP 20034	NetBus 2 Pro
TCP 20432	Shaft intruder-to-master
UDP 20433	Shaft daemon-to-master
TCP 27374	SubSeven server (default for V2.1-Defcon)
UDP 27444	Trinoo master-to-daemon
TCP 27665	Trinoo intruder-to-master
TCP 30100	NetSphere
UDP 31335	Trinoo daemon-to-master
TCP 31337	Back Orifice, Baron Night, Bo Facil
TCP 33270	Trinity master-to-daemon
TCP 33567	Backdoor rootshell via inetd (from Lion worm)

TCP 33568	Trojaned version of SSH (from Lion worm)
TCP 40421	Masters Paradise Trojan horse
TCP 60008	Backdoor rootshel via inetd (from Lion worm)
TCP 65000	Stacheldraht master-to-daemon

Table 8. Commonly affected Bad/Vulnerable ports

Each port is assigned a number from 0 to 65535. Here, there are three different types of numbers that need to be taken into account with one another:

- The ports 0 to 1023 belong to the standardized ports, which the Internet Assigned Numbers Authority (IANA) are mostly responsible for assigning. Following this, the port 80 is reserved for HTTP connections and for this reason is the most important port for web server requests.
- The port numbers 1024 to 49151 are reserved for registered services by default. However, these are also assigned to client programs, especially when it comes to Linux systems.
- The ports 49152 to 65535 dynamically assign operating systems to clients.

In order to establish a connection via a certain port, this has to be opened, i.e. activated. Regarding online data transfer, this means having a high number of open ports, which carries with it a certain number or risks: if their respective applications contain security gaps, each open port presents a potential access point for attackers. For this reason, it's important to always keep an eye on which ports are open on the system and which applications are operating behind this running flow of data.

10. CONCLUSION AND FUTURE WORK

With the increased rate of cyber-attacks, Penetration testing is considered one of the most famous and preventive testing process to secure the system from vulnerabilities or future cyber-attacks. By understanding and finding the weaknesses of the system we can prevent the gateway for cyber-attacks and secure the system in a stronger way.

Throughout my thesis period, I became aware of various benefits of this process and as discussed in chapter 7 the comparison between the existing tools helps to understand the features of various tools and became aware of demerits of existing tools. As a result I am now aware that most of tools are not completely satisfying what they claim in their website as their features and scanning for same vulnerabilities. Except few, many tools are not providing detailed information which will be hard for a beginner to understand the results.

Passwords can be considered an appropriate safeguard to ensure the security of accounts and the confidentiality of sensitive information, provided an appropriate GDPR password policy is in place, my implementation of secure and strong password checking tool will be helpful to create a strong password In order to maintain security and to prevent processing in infringement of the GDPR. By implementing Application scanning tool which includes features that are not implemented by many available tools I am able to understand the importance of ports and their important role in security.

In Future I am looking forward to enhance my Application scanning tool by combining secure login system and including more scanning information by checking for buffer overflow, XML injection and validating WCMS.

References

[1] P. Prasad, Mastering Modern Web Penetration Testing, Packt Publishing, 2016, ISBN: 978-1785284588

[2] J. A. Ansari, Web Penetration Testing with Kali Linux - Second Edition, Packt Publishing 2015, ISBN: 978-1783988525

[3] OWASP Zed Attack Proxy Project [online] - www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

[4] Acunetix web vulnerability scanner[online] - https://www.acunetix.com

[5] Vega open source web security scanner - https://subgraph.com/vega

[6] Pentest-tools.com [online] web application pentest tool- https://pentest-tools.com/home

[7] GDPR - https://eugdpr.org/

[8] Testing for web applications [online] - https://www.testbytes.net/blog/security-testing-for-web-applications

[9] OWASP cheat sheet - https://www.owasp.org/index.php/Web_Application_Security_Testing_Cheat_Sheet

[10] Penetration test [online] - https://en.wikipedia.org/wiki/Penetration_test

[11] Burp Suit [online] - https://www.pentestgeek.com/what-is-burpsuite

[12] Metasploit [online] - https://www.metasploit.com/

[13] Nmap [online] - https://nmap.org/

[14] TCP [online] - https://en.wikipedia.org/wiki/Transmission_Control_Protocol

[15] List of TCP and UDP port numbers[online] - https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

[16] How penetration testing can help you comply with the GDPR - https://www.itgovernance.co.uk/blog/how-penetration-testing-can-help-you-comply-with-the-gdpr

[17] The Types of Penetration Testing - https://resources.infosecinstitute.com/the-types-of-penetration-testing/#gref

[18] The History of Penetration Testing -

https://resources.infosecinstitute.com/category/certifications-training/pentesting-certifications/pentesting-history/#gref

[19] w3af [online] - http://w3af.org/

[20] An Overview of Penetration Testing [offline] -

https://www.researchgate.net/publication/274174058_An_Overview_of_Penetration_Testing

[21] Eugene Spafford, An Information Security Pioneer, IEEE publishing, 2008, **ISSN:** 1558-4046

[22] Ethical Hacking and Penetration Testing Guide - http://www.lepointdeau.fr/Ethical%20Hacking%20and%20Penetration%20Testing%20Guide%20-%20Baloch,%20Rafay.pdf

[23] SQL injection - https://en.wikipedia.org/wiki/SQL_injection

[24] Cross-site Scripting (XSS) -

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

[25] NIST Special Publication 800-63B guidelines -

https://pages.nist.gov/800-63-3/sp800-63b.html#sec5